

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Application Program Interface for Network Software  
Platform**

Inventor(s):  
Brad Abrams

ATTORNEY'S DOCKET NO. MS1-863US

1 **TECHNICAL FIELD**

2 This invention relates to network software, such as Web applications, and to  
3 computer software development of such network software. More particularly, this  
4 invention relates to an application program interface (API) that facilitates use of a  
5 network software platform by application programs and computer hardware.  
6

7 **BACKGROUND**

8 Very early on, computer software came to be categorized as "operating  
9 system" software or "application" software. Broadly speaking, an application is  
10 software meant to perform a specific task for the computer user such as solving a  
11 mathematical equation or supporting word processing. The operating system is  
12 the software that manages and controls the computer hardware. The goal of the  
13 operating system is to make the computer resources available to the application  
14 programmer while at the same time, hiding the complexity necessary to actually  
15 control the hardware.

16 The operating system makes the resources available via functions that are  
17 collectively known as the Application Program Interface or API. The term API is  
18 also used in reference to a single one of these functions. The functions are often  
19 grouped in terms of what resource or service they provide to the application  
20 programmer. Application software requests resources by calling individual API  
21 functions. API functions also serve as the means by which messages and  
22 information provided by the operating system are relayed back to the application  
23 software.

24 In addition to changes in hardware, another factor driving the evolution of  
25 operating system software has been the desire to simplify and speed application



1 software development. Application software development can be a daunting task,  
2 sometimes requiring years of developer time to create a sophisticated program  
3 with millions of lines of code. For a popular operating system such as Microsoft  
4 Windows®, application software developers write thousands of different  
5 applications each year that utilize the operating system. A coherent and usable  
6 operating system base is required to support so many diverse application  
7 developers.

8 Often, development of application software can be made simpler by making  
9 the operating system more complex. That is, if a function may be useful to several  
10 different application programs, it may be better to write it once for inclusion in the  
11 operating system, than requiring dozens of software developers to write it dozens  
12 of times for inclusion in dozens of different applications. In this manner, if the  
13 operating system supports a wide range of common functionality required by a  
14 number of applications, significant savings in applications software development  
15 costs and time can be achieved.

16 Regardless of where the line between operating system and application  
17 software is drawn, it is clear that for a useful operating system, the API between  
18 the operating system and the computer hardware and application software is as  
19 important as efficient internal operation of the operating system itself.

20 Over the past few years, the universal adoption of the Internet, and  
21 networking technology in general, has changed the landscape for computer  
22 software developers. Traditionally, software developers focused on single-site  
23 software applications for standalone desktop computers, or LAN-based computers  
24 that were connected to a limited number of other computers via a local area  
25 network (LAN). Such software applications were typically referred to as “shrink

1 wrapped" products because the software was marketed and sold in a shrink-  
2 wrapped package. The applications utilized well-defined APIs to access the  
3 underlying operating system of the computer.

4 As the Internet evolved and gained widespread acceptance, the industry  
5 began to recognize the power of hosting applications at various sites on the World  
6 Wide Web (or simply the "Web"). In the networked world, clients from anywhere  
7 could submit requests to server-based applications hosted at diverse locations and  
8 receive responses back in fractions of a second. These Web applications, however,  
9 were typically developed using the same operating system platform that was  
10 originally developed for standalone computing machines or locally networked  
11 computers. Unfortunately, in some instances, these applications do not adequately  
12 transfer to the distributed computing regime. The underlying platform was simply  
13 not constructed with the idea of supporting limitless numbers of interconnected  
14 computers.

15 To accommodate the shift to the distributed computing environment being  
16 ushered in by the Internet, Microsoft Corporation is developing a network  
17 software platform known as the ".NET" platform (read as "Dot Net"). The  
18 platform allows developers to create Web services that will execute over the  
19 Internet. Such a dynamic shift requires a new ground-up design of an entirely new  
20 API.

21 In response to this challenge, the inventors developed a unique set of API  
22 functions for Microsoft's .NET™ platform.  
23  
24  
25

## **SUMMARY**

An application program interface (API) provides a set of functions for application developers who build Web applications on a network platform, such as Microsoft Corporation's .NET™ platform.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

The same numbers are used throughout the drawings to reference like features.

Fig. 1 illustrates a network architecture in which clients access Web services over the Internet using conventional protocols.

Fig. 2 is a block diagram of a software architecture for Microsoft's .NET™ platform, which includes an application program interface (API).

Fig. 3 is a block diagram of unique namespaces supported by the API, as well as function classes of the various API functions.

Fig. 4 is a block diagram of an exemplary computer that may execute all or part of the software architecture.

## **BRIEF DESCRIPTION OF ACCOMPANYING COMPACT DISC**

Accompanying this specification is a compact disc that stores a compiled HTML help file identifying the API (application program interface) for Microsoft's .NET™ network platform. The file is named "cpref.chm" and was created on June 8, 2001. It is 30.81 Mbytes in size. The file can be executed on a Windows®-based computing device (e.g., IBM-PC, or equivalent) that executes a Windows®-brand operating system (e.g., Windows® NT, Windows® 98,

1 Windows® 2000, etc.). The compiled HTML help file stored on the compact disk  
2 is hereby incorporated by reference.

3 Additionally, the APIs contained in the compiled HTML help file are also  
4 provided in approximately 100 separate text files named "NamespaceName.txt".  
5 The text files comply with the ASCII format.

6 The compact disc itself is a CD-ROM, and conforms to the ISO 9660  
7 standard.

### 8 9 **DETAILED DESCRIPTION**

10 This disclosure addresses an application program interface (API) for a  
11 network platform upon which developers can build Web applications and services.  
12 More particularly, an exemplary API is described for the .NET™ platform created  
13 by Microsoft Corporation. The .NET™ platform is a software platform for Web  
14 services and Web applications implemented in the distributed computing  
15 environment. It represents the next generation of Internet computing, using open  
16 communication standards to communicate among loosely coupled Web services  
17 that are collaborating to perform a particular task.

18 In the described implementation, the .NET™ platform utilizes XML  
19 (extensible markup language), an open standard for describing data. XML is  
20 managed by the World Wide Web Consortium (W3C). XML is used for defining  
21 data elements on a Web page and business-to-business documents. XML uses a  
22 similar tag structure as HTML; however, whereas HTML defines how elements  
23 are displayed, XML defines what those elements contain. HTML uses predefined  
24 tags, but XML allows tags to be defined by the developer of the page. Thus,  
25 virtually any data items can be identified, allowing Web pages to function like

1 database records. Through the use of XML and other open protocols, such as  
2 Simple Object Access Protocol (SOAP), the .NET™ platform allows integration of  
3 a wide range of services that can be tailored to the needs of the user. Although the  
4 embodiments described herein are described in conjunction with XML and other  
5 open standards, such are not required for the operation of the claimed invention.  
6 Other equally viable technologies will suffice to implement the inventions  
7 described herein.

8 As used herein, the phrase application program interface or API includes  
9 traditional interfaces that employ method or function calls, as well as remote calls  
10 (e.g., a proxy, stub relationship) and SOAP/XML invocations.

#### 12 EXEMPLARY NETWORK ENVIRONMENT

13 Fig. 1 shows a network environment 100 in which a network platform, such  
14 as the .NET™ platform, may be implemented. The network environment 100  
15 includes representative Web services 102(1), ..., 102(N), which provide services  
16 that can be accessed over a network 104 (e.g., Internet). The Web services,  
17 referenced generally as number 102, are programmable application components  
18 that are reusable and interact programmatically over the network 104, typically  
19 through industry standard Web protocols, such as XML, SOAP, WAP (wireless  
20 application protocol), HTTP (hypertext transport protocol), and SMTP (simple  
21 mail transfer protocol) although other means of interacting with the Web services  
22 over the network may also be used, such as Remote Procedure Call (RPC) or  
23 object broker type technology. A Web service can be self-describing and is often  
24 defined in terms of formats and ordering of messages.

1 Web services 102 are accessible directly by other services (as represented  
2 by communication link 106) or a software application, such as Web application  
3 110 (as represented by communication links 112 and 114). Each Web service 102  
4 is illustrated as including one or more servers that execute software to handle  
5 requests for particular services. Such services often maintain databases that store  
6 information to be served back to requesters. Web services may be configured to  
7 perform any one of a variety of different services. Examples of Web services  
8 include login verification, notification, database storage, stock quoting, location  
9 directories, mapping, music, electronic wallet, calendar/scheduler, telephone  
10 listings, news and information, games, ticketing, and so on. The Web services can  
11 be combined with each other and with other applications to build intelligent  
12 interactive experiences.

13 The network environment 100 also includes representative client devices  
14 120(1), 120(2), 120(3), 120(4), ..., 120(M) that utilize the Web services 102 (as  
15 represented by communication link 122) and/or the Web application 110 (as  
16 represented by communication links 124, 126, and 128). The clients may  
17 communicate with one another using standard protocols as well, as represented by  
18 an exemplary XML link 130 between clients 120(3) and 120(4).

19 The client devices, referenced generally as number 120, can be  
20 implemented many different ways. Examples of possible client implementations  
21 include, without limitation, portable computers, stationary computers, tablet PCs,  
22 televisions/set-top boxes, wireless communication devices, personal digital  
23 assistants, gaming consoles, printers, photocopiers, and other smart devices.

24 The Web application 110 is an application designed to run on the network  
25 platform and may utilize the Web services 102 when handling and servicing

1 requests from clients 120. The Web application 110 is composed of one or more  
2 software applications 130 that run atop a programming framework 132, which are  
3 executing on one or more servers 134 or other computer systems. Note that a  
4 portion of Web application 110 may actually reside on one or more of clients 120.  
5 Alternatively, Web application 110 may coordinate with other software on clients  
6 120 to actually accomplish its tasks.

7 The programming framework 132 is the structure that supports the  
8 applications and services developed by application developers. It permits multi-  
9 language development and seamless integration by supporting multiple languages.  
10 It supports open protocols, such as SOAP, and encapsulates the underlying  
11 operating system and object model services. The framework provides a robust and  
12 secure execution environment for the multiple programming languages and offers  
13 secure, integrated class libraries.

14 The framework 132 is a multi-tiered architecture that includes an  
15 application program interface (API) layer 142, a common language runtime (CLR)  
16 layer 144, and an operating system/services layer 146. This layered architecture  
17 allows updates and modifications to various layers without impacting other  
18 portions of the framework. A common language specification (CLS) 140 allows  
19 designers of various languages to write code that is able to access underlying  
20 library functionality. The specification 140 functions as a contract between  
21 language designers and library designers that can be used to promote language  
22 interoperability. By adhering to the CLS, libraries written in one language can be  
23 directly accessible to code modules written in other languages to achieve seamless  
24 integration between code modules written in one language and code modules  
25 written in another language. One exemplary detailed implementation of a CLS is

described in an ECMA standard created by participants in ECMA TC39/TG3.

The reader is directed to the ECMA web site at [www.ecma.ch](http://www.ecma.ch).

The API layer 142 presents groups of functions that the applications 130 can call to access the resources and services provided by layer 146. By exposing the API functions for a network platform, application developers can create Web applications for distributed computing systems that make full use of the network resources and other Web services, without needing to understand the complex interworkings of how those network resources actually operate or are made available. Moreover, the Web applications can be written in any number of programming languages, and translated into an intermediate language supported by the common language runtime 144 and included as part of the common language specification 140. . In this way, the API layer 142 can provide methods for a wide and diverse variety of applications.

Additionally, the framework 132 can be configured to support API calls placed by remote applications executing remotely from the servers 134 that host the framework. Representative applications 148(1) and 148(2) residing on clients 120(3) and 120(M), respectively, can use the API functions by making calls directly, or indirectly, to the API layer 142 over the network 104.

The framework may also be implemented at the clients. Client 120(3) represents the situation where a framework 150 is implemented at the client. This framework may be identical to server-based framework 132, or modified for client purposes. Alternatively, the client-based framework may be condensed in the event that the client is a limited or dedicated function device, such as a cellular phone, personal digital assistant, handheld computer, or other communication/computing device.



## DEVELOPERS' PROGRAMMING FRAMEWORK

Fig. 2 shows the programming framework 132 in more detail. The common language specification (CLS) layer 140 supports applications written in a variety of languages 130(1), 130(2), 130(3), 130(4), ..., 130(K). Such application languages include Visual Basic, C++, C#, COBOL, Jscript, Perl, Eiffel, Python, and so on. The common language specification 140 specifies a subset of features or rules about features that, if followed, allow the various languages to communicate. For example, some languages do not support a given type (e.g., an "int\*" type) that might otherwise be supported by the common language runtime 144. In this case, the common language specification 140 does not include the type. On the other hand, types that are supported by all or most languages (e.g., the "int[]" type) is included in common language specification 140 so library developers are free to use it and are assured that the languages can handle it. This ability to communicate results in seamless integration between code modules written in one language and code modules written in another language. Since different languages are particularly well suited to particular tasks, the seamless integration between languages allows a developer to select a particular language for a particular code module with the ability to use that code module with modules written in different languages. The common language runtime 144 allow seamless multi-language development, with cross language inheritance, and provide a robust and secure execution environment for the multiple programming languages. For more information on the common language specification 140 and the common language runtime 144, the reader is directed to co-pending applications entitled "Method and System for Compiling Multiple Languages", filed 6/21/2000 (serial

1 number 09/598,105) and "Unified Data Type System and Method" filed 7/10/2000  
2 (serial number 09/613,289), which are incorporated by reference.

3 The framework 132 encapsulates the operating system 146(1) (e.g.,  
4 Windows®-brand operating systems) and object model services 146(2) (e.g.,  
5 Component Object Model (COM) or Distributed COM). The operating system  
6 146(1) provides conventional functions, such as file management, notification,  
7 event handling, user interfaces (e.g., windowing, menus, dialogs, etc.), security,  
8 authentication, verification, processes and threads, memory management, and so  
9 on. The object model services 146(2) provide interfacing with other objects to  
10 perform various tasks. Calls made to the API layer 142 are handed to the common  
11 language runtime layer 144 for local execution by the operating system 146(1)  
12 and/or object model services 146(2).

13 The API 142 groups API functions into multiple namespaces. Namespaces  
14 essentially define a collection of classes, interfaces, delegates, enumerations, and  
15 structures, which are collectively called "types", that provide a specific set of  
16 related functionality. A class represents managed heap allocated data that has  
17 reference assignment semantics. A delegate is an object oriented function pointer.  
18 An enumeration is a special kind of value type that represents named constants. A  
19 structure represents static allocated data that has value assignment semantics. An  
20 interface defines a contract that other types can implement.

21 By using namespaces, a designer can organize a set of types into a  
22 hierarchical namespace. The designer is able to create multiple groups from the  
23 set of types, with each group containing at least one type that exposes logically  
24 related functionality. In the exemplary implementation, the API 142 is organized  
25 into four root namespaces: a first namespace 200 for Web applications, a second

namespace 202 for client applications, a third namespace 204 for data and XML, and a fourth namespace 206 for base class libraries (BCLs). Each group can then be assigned a name. For instance, types in the Web applications namespace 200 are assigned the name "Web", and types in the data and XML namespace 204 can be assigned names "Data" and "XML" respectively. The named groups can be organized under a single "global root" namespace for system level APIs, such as an overall System namespace. By selecting and prefixing a top level identifier, the types in each group can be easily referenced by a hierarchical name that includes the selected top level identifier prefixed to the name of the group containing the type. For instance, types in the Web applications namespace 200 can be referenced using the hierarchical name "System.Web". In this way, the individual namespaces 200, 202, 204, and 206 become major branches off of the System namespace and can carry a designation where the individual namespaces are prefixed with a designator, such as a "System." prefix.

The Web applications namespace 200 pertains to Web based functionality, such as dynamically generated Web pages (e.g., Microsoft's Active Server Pages (ASP)). It supplies types that enable browser/server communication. The client applications namespace 202 pertains to drawing and client side UI functionality. It supplies types that enable drawing of two-dimensional (2D), imaging, and printing, as well as the ability to construct window forms, menus, boxes, and so on.

The data and XML namespace 204 relates to connectivity to data sources and XML functionality. It supplies classes, interfaces, delegates, and enumerations that enable security, specify data types, and serialize objects into XML format documents or streams. The base class libraries (BCL) namespace

206 pertains to basic system and runtime functionality. It contains the fundamental types and base classes that define commonly-used value and reference data types, events and event handlers, interfaces, attributes, and processing exceptions.

In addition to the framework 132, programming tools 210 are provided to assist the developer in building Web services and/or applications. One example of the programming tools 200 is Visual Studio™, a multi-language suite of programming tools offered by Microsoft Corporation.

#### ROOT API NAMESPACES

Fig. 3 shows the API 142 and its four root namespaces in more detail. In one embodiment, the namespaces are identified according to a hierarchical naming convention in which strings of names are concatenated with periods. For instance, the Web applications namespace 200 is identified by the root name "System.Web". Within the "System.Web" namespace is another namespace for Web services, identified as "System.Web.Services", which further identifies another namespace for a description known as "System.Web.Services.Description". With this naming convention in mind, the following provides a general overview of selected namespaces of the API 142, although other naming conventions could be used with equal effect.

The Web applications namespace 200 ("System.Web") defines additional namespaces, including:

- A services namespace 300 ("System.Web.Services") containing classes that enable a developer to build and use Web services. The

services namespace 300 defines additional namespaces, including a description namespace 302 ("System.Web.Services.Description") containing classes that enable a developer to publicly describe a Web service via a service description language (such as WSDL, a specification available from the W3C), a discovery namespace 304 ("System.Web.Services.Discovery") containing classes that allow Web service consumers to locate available Web Services on a Web server, and a protocols namespace 306 ("System.Web.Services.Protocols") containing classes that define the protocols used to transmit data across a network during communication between Web service clients and the Web service itself.

- A caching namespace 308 ("System.Web.Caching") containing classes that enable developers to decrease Web application response time through temporarily caching frequently used resources on the server. This includes ASP.NET pages, web services, and user controls. (ASP.NET is the updated version of Microsoft's ASP technology.) Additionally, a cache dictionary is available for developers to store frequently used resources, such as hash tables and other data structures.
- A configuration namespace 310 ("System.Web.Configuration") containing classes that are used to read configuration data in for an application.
- A UI namespace 312 ("System.Web.UI") containing types that allow developers to create controls and pages that will appear in Web

1 applications as user interfaces on a Web page. This namespace  
2 includes the control class, which provides all web based controls,  
3 whether those encapsulating HTML elements, higher level Web  
4 controls, or even custom User controls, with a common set of  
5 functionality. Also provided are classes which provide the web  
6 forms server controls data binding functionality, the ability to save  
7 the view state of a given control or page, as well as parsing  
8 functionality for both programmable and literal controls. Within the  
9 UI namespace 312 are two additional namespaces: an HTML  
10 controls namespace 314 ("System.Web.UI.HtmlControls")  
11 containing classes that permit developers to interact with types that  
12 encapsulates html 3.2 elements create HTML controls, and a Web  
13 controls namespace 316 ("System.Web.UI.WebControls")  
14 containing classes that allow developers to create higher level Web  
15 controls.

- 16 • A security namespace 318 ("System.Web.Security") containing  
17 classes used to implement security in web server applications, such  
18 as basic authentication, challenge response authentication, and role  
19 based authentication.
- 20 • A session state namespace 320 ("System.Web.SessionState")  
21 containing classes used to access session state values (i.e., data that  
22 lives across requests for the lifetime of the session) as well as  
23 session-level settings and lifetime management methods.

24  
25 The client applications namespace 202 is composed of two namespaces:

- A windows forms namespace 322 (“System.Windows.Forms”) containing classes for creating Windows®-based client applications that take full advantage of the rich user interface features available in the Microsoft Windows® operating system, such as the ability to drag and drop screen elements. Such classes may include wrapped APIs available in the Microsoft Windows® operating system that are used in a windowing UI environment. Within this namespace are a design namespace 324 (“System.Windows.Forms.Design”) that contains classes to extend design-time support for Windows forms and a component model namespace 326 (“System.Windows.Forms.ComponentModel”) that contains the windows form implementation of the general component model defined in System.ComponentModel. This namespace contains designer tools, such as Visual Studio, which offer a rich experience for developers at design time.
- A drawing namespace 328 (“System.Drawing”) containing classes for graphics functionality. The drawing namespace 328 includes a 2D drawing namespace 330 (“System.Drawing.Drawing2D”) that contains classes and enumerations to provide advanced 2-dimensional and vector graphics functionality, an imaging namespace 332 (“System.Drawing.Imaging”) that contains classes for advanced imaging functionality, a printing namespace 334 (“System.Drawing.Printing”) that contains classes to permit developers to customize printing, and a text namespace 336

1 ("System.Drawing.Text") that contains classes for advanced  
2 typography functionality.

3  
4 The data and XML namespace 204 is composed of two namespaces:

- 5
- 6 • A data namespace 340 ("System.Data") containing classes that  
7 enable developers to build components that efficiently manage data  
8 from multiple data sources. It implements an architecture that, in a  
9 disconnected scenario (such as the Internet), provides tools to  
10 request, update, and reconcile data in multiple tier systems. The data  
11 namespace 340 includes a common namespace 342 that contains  
12 types shared by data providers. A data provider describes a  
13 collection of types used to access a data source, such as a database,  
14 in the managed space. The data namespace 340 also includes an  
15 OLE DB namespace 344 that contains types pertaining to data used  
16 in object-oriented databases (e.g., Microsoft's SQL Server), and a  
17 SQL client namespace 346 that contains types pertaining to data  
18 used by SQL clients. The data namespace also includes a SQL types  
19 namespace 348 ("System.Data.SqlTypes") that contains classes for  
20 native data types within Microsoft's SQL Server. The classes  
21 provide a safer, faster alternative to other data types. Using the  
22 objects within this namespace helps prevent type conversion errors  
23 caused in situations where loss of precision could occur. Because  
24 other data types are converted to and from SQL types behind the  
25



scenes, explicitly creating and using objects within this namespace results in faster code as well.

- An XML namespace 350 (“System.XML”) containing classes that provide standards-based support for processing XML. The supported standards include XML (e.g., version 1.0), XML Namespaces (both stream level and DOM), XML Schemas, XPath expressions, XSL/T transformations, DOM Level 2 Core, and SOAP (e.g., version 1.1). The XML namespace 350 includes an XSLT namespace 352 (“System.XML.Xsl”) that contains classes and enumerations to support XSLT (Extensible Stylesheet Language Transformations), an XPath namespace 354 (“System.XML.XPath”) that contains an XPath parser and evaluation engine, and a serialization namespace 356 (“System.XML.Serialization”) that contains classes used to serialize objects into XML format documents or streams.

The base class library namespace 206 (“System”) includes the following namespaces:

- A collections namespace 360 (“System.Collections”) containing interfaces and classes that define various collections of objects, such as lists, queues, arrays, hash tables and dictionaries.
- A configuration namespace 362 (“System.Configuration”) containing classes and interfaces that allow developers to programmatically access configuration settings and handle errors in configuration files.

- A diagnostics namespace 364 (“System.Diagnostics”) containing classes that are used to debug applications and to trace code execution. The namespace allows developers to start system processes, read and write to event logs, and monitor system performance using performance counters.
- A globalization namespace 366 (“System.Globalization”) containing classes that define culture-related information, including the language, the country/region, the calendars in use, the format patterns for dates, currency and numbers, and the sort order for strings.
- An I/O namespace 368 (“System.IO”) containing the infrastructure pieces to operate with the input/output of data streams, files, and directories. This namespace includes a model for working with streams of bytes, higher level readers and writers which consume those bytes, various constructions or implementations of the streams (e.g., FileStream and MemoryStream) and, a set of utility classes for working with files and directories.
- A net namespace 370 (“System.Net”) providing an extensive set of classes for building network-enabled application, referred to as the Net Class Libraries (NCL). One element to the design of the Net Class Libraries is an extensible, layered approach to exposing networking functionality. The NCL stack contains three basic layers. A base layer (System.Net.Socket) provides access to an interface to TCP/IP, the communications protocol of UNIX networks and the Internet. One example of such an interface is the “WinSock

1 API" from Microsoft Corporation. The next layer is the Transport  
2 Protocol classes, which support such transport protocols as TCP and  
3 UDP. Developers may write their own protocol classes to provide  
4 support for protocols such as IGMP and ICMP. The third layer is  
5 the Web request, which provides an abstract factory pattern for the  
6 creation of other protocol classes. The NCL provides  
7 implementations for Hyper Text Transport Protocol (HTTP).

- 8 • A reflection namespace ("System.Reflection") 372 containing types  
9 that provide a managed view of loaded types, methods, and fields,  
10 with the ability to dynamically create and invoke types.
- 11 • A resources namespace 374 ("System.Resources") containing  
12 classes and interfaces that allow developers to create, store and  
13 manage various culture-specific resources used in an application.
- 14 • A security namespace 376 ("System.Security") supporting the  
15 underlying structure of the security system, including interfaces,  
16 attributes, exceptions, and base classes for permissions.
- 17 • A service process namespace 378 ("System.ServiceProcess")  
18 containing classes that allow developers to install and run services.  
19 Services are long-running executables that run without a user  
20 interface. They can be installed to run under a system account that  
21 enables them to be started at computer reboot. Services whose  
22 implementation is derived from processing in one class can define  
23 specific behavior for start, stop, pause, and continue commands, as  
24 well as behavior to take when the system shuts down.

- A text namespace 380 (“System.Text”) containing classes representing various types of encodings (e.g., ASCII, Unicode, UTF-7, and UTF-8), abstract base classes for converting blocks of characters to and from blocks of bytes, and a helper class that manipulates and formats string objects without creating intermediate instances.
- A threading namespace 382 (“System.Threading”) containing classes and interfaces that enable multi-threaded programming. The threading namespace includes a ThreadPool class that manages groups of threads, a Timer class that enables a delegate to be called after a specified amount of time, and a Mutex class for synchronizing mutually-exclusive threads. This namespace also provides classes for thread scheduling, wait notification, and deadlock resolution.
- A runtime namespace 384 (“System.Runtime”) containing multiple namespaces concerning runtime features, including an interoperation services namespace 386 (“System.Runtime.InteropServices”) that contains a collection of classes useful for accessing COM objects. The types in the InteropServices namespace fall into the following areas of functionality: attributes, exceptions, managed definitions of COM types, wrappers, type converters, and the Marshal class. The runtime namespace 384 further includes a remoting namespace 388 (“System.Runtime.Remoting”) that contains classes and interfaces allowing developers to create and configure distributed applications. Another namespace within the runtime namespace 384 is a

1 serialization namespace 390 ("System.Runtime.Serialization") that  
2 contains classes used for serializing and deserializing objects.  
3 Serialization is the process of converting an object or a graph of  
4 objects into a linear sequence of bytes for either storage or  
5 transmission to another location.

6  
7  
8 The web applications namespace 200 ("System.Web") defines several  
9 additional namespaces, including the services namespace 300  
10 ("System.Web.Services"), a caching namespace 308 ("System.Web.Caching"), a  
11 configuration namespace 310 ("System.Web.Configuration"), a UI namespace 312  
12 ("System.Web.UI"), a security namespace 318 ("System.Web.Security"), and a  
13 session state namespace 320 ("System.Web.SessionState"). In general, the web  
14 applications namespace 200 supplies tools that enable browser-server  
15 communication.

16 The services namespace 300 contains classes that allow developers to build  
17 and use various web services. The services namespace includes a web service  
18 class that defines a base class for web services and a web method attribute class  
19 that allows a method to be programmatically exposed over the web.

20 The UI namespace 312 contains classes that allow a user to create HTML  
21 server controls on a web page. These HTML server controls execute on the server  
22 and map to standard HTML tags. The UI namespace also contains classes that  
23 allow a user to create web server controls on a web page. These web server  
24 controls run on the web server and include form controls, such as buttons and text  
25 boxes.

1 The web applications namespace 200 also includes classes for manipulating  
2 cookies, transferring files, handling exception information, and controlling an  
3 output cache. Specific details regarding the System.Web namespace are provided  
4 below.

## 6 System.Web

### 7 *Description*

8 The System.Web namespace supplies classes and interfaces that enable  
9 browser/server communication. This namespace includes the HttpRequest class  
10 that provides extensive information about the current HTTP request, the  
11 HttpResponse class that manages HTTP output to the client, and the  
12 HttpServerUtility object that provides access to server-side utilities and  
13 processes. System.Web also includes classes for cookie manipulation, file transfer,  
14 exception information, and output cache control.

15 BeginEventHandler delegate (System.Web)

### 18 *Description*

20 EndEventHandler delegate (System.Web)

### 23 *Description*

25 HttpWorkerRequest.EndOfSendNotification delegate (System.Web)

1  
2  
3 *Description*

4  
5 HttpApplication class (System.Web)

6  
7  
8 *Description*

9 Defines the methods, properties, and events common to all application  
10 objects within an ASP.NET application.

11 Constructors:

12 HttpApplication

13 *Example Syntax:*

14  
15 [C#] public HttpApplication();

16 [C++] public: HttpApplication();

17 [VB] Public Sub New()

18 [JScript] public function HttpApplication();

19 Properties:

20 Application

21  
22 [C#] public HttpSessionState Application {get;}

23 [C++] public: \_\_property HttpSessionState\* get\_Application();

24 [VB] Public ReadOnly Property Application As HttpSessionState

25 [JScript] public function get Application() : HttpSessionState;

1  
2 *Description*

3 Gets a reference to an **HTTPApplication** state bag instance.

4 Context

5  
6 [C#] public HttpContext Context {get;}  
7 [C++] public: \_\_property HttpContext\* get\_Context();  
8 [VB] Public ReadOnly Property Context As HttpContext  
9 [JScript] public function get Context() : HttpContext;  
10

11 *Description*

12 Gets the **HTTPRuntime** -provided context object that provides access to  
13 additional pipeline-module exposed objects.

14 Events

15  
16 [C#] protected EventHandlerList Events {get;}  
17 [C++] protected: \_\_property EventHandlerList\* get\_Events();  
18 [VB] Protected ReadOnly Property Events As EventHandlerList  
19 [JScript] protected function get Events() : EventHandlerList;  
20

21 *Description*

22  
23 Modules

24  
25 [C#] public HttpModuleCollection Modules {get;}



```

1  [C++]    public:    __property    HttpModuleCollection*    get_Modules();
2  [VB]    Public    ReadOnly    Property    Modules    As    HttpModuleCollection
3  [JScript]    public    function    get    Modules()    :    HttpModuleCollection;
4

```

#### *Description*

Gets the collection of **HTTPModules** configured for the current application.

#### *Request*

```

10 [C#]      public      HttpRequest      Request      {get;}
11 [C++]     public:     __property     HttpRequest*     get_Request();
12 [VB]     Public     ReadOnly     Property     Request     As     HttpRequest
13 [JScript]     public     function     get     Request()     :     HttpRequest;
14

```

#### *Description*

Gets the intrinsic object that provides access to incoming **HttpRequest** data.

#### *Response*

```

20 [C#]      public      HttpResponse      Response      {get;}
21 [C++]     public:     __property     HttpResponse*     get_Response();
22 [VB]     Public     ReadOnly     Property     Response     As     HttpResponse
23 [JScript]     public     function     get     Response()     :     HttpResponse;
24

```

#### *Description*

1 The intrinsic object that allows transmission of **HttpResponse** data to a  
2 client.

3 Server

4  
5 [C#] public HttpServerUtility Server {get;}

6 [C++] public: \_\_property HttpServerUtility\* get\_Server();

7 [VB] Public ReadOnly Property Server As HttpServerUtility

8 [JScript] public function get Server() : HttpServerUtility;

9  
10 *Description*

11 Gets the intrinsic **Server** object.

12 Session

13  
14 [C#] public HttpSessionState Session {get;}

15 [C++] public: \_\_property HttpSessionState\* get\_Session();

16 [VB] Public ReadOnly Property Session As HttpSessionState

17 [JScript] public function get Session() : HttpSessionState;

18  
19 *Description*

20 Gets the intrinsic **Session** object that provides access to session data.

21 Site

22  
23 [C#] public ISite Site {get; set;}

24 [C++] public: \_\_property ISite\* get\_Site();public: \_\_property void

25 set\_Site(ISite\*);

```

1  [VB]          Public          Property          Site          As          ISite
2  [JScript] public function get Site() : ISite;public function set Site(ISite);
3

```

4 *Description*

6 User

```

8  [C#]          public          IPrincipal          User          {get;}
9  [C++]          public:          __property          IPrincipal*          get_User();

```

```

10 [VB] Public ReadOnly Property User As IPrincipal

```

```

11 [JScript] public function get User() : IPrincipal;

```

13 *Description*

14 Gets the User intrinsic object.

```

16 [C#] public event EventHandler AcquireRequestState;

```

```

17 [C++] public: __event EventHandler* AcquireRequestState;

```

```

18 [VB] Public Event AcquireRequestState As EventHandler

```

20 *Description*

```

23 [C#] public event EventHandler AuthenticateRequest;

```

```

24 [C++] public: __event EventHandler* AuthenticateRequest;

```

```

25 [VB] Public Event AuthenticateRequest As EventHandler

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

*Description*

[C#]	public	event	EventHandler	AuthorizeRequest;
[C++]	public:	__event	EventHandler*	AuthorizeRequest;
[VB]	Public	Event	AuthorizeRequest	As EventHandler

*Description*

[C#]	public	event	EventHandler	BeginRequest;
[C++]	public:	__event	EventHandler*	BeginRequest;
[VB]	Public	Event	BeginRequest	As EventHandler

*Description*

[C#]	public	event	EventHandler	Disposed;	
[C++]	public:	__sealed	__event	EventHandler*	Disposed;
[VB]	NotOverridable	Public	Event	Disposed	As EventHandler

*Description*

TOP SECRET

1  
2 [C#] public event EventHandler EndRequest;  
3 [C++] public: \_\_event EventHandler\* EndRequest;  
4 [VB] Public Event EndRequest As EventHandler

5  
6 *Description*

7  
8  
9 [C#] public event EventHandler Error;  
10 [C++] public: \_\_event EventHandler\* Error;  
11 [VB] Public Event Error As EventHandler

12  
13 *Description*

14  
15  
16 [C#] public event EventHandler PostRequestHandlerExecute;  
17 [C++] public: \_\_event EventHandler\* PostRequestHandlerExecute;  
18 [VB] Public Event PostRequestHandlerExecute As EventHandler

19  
20 *Description*

21  
22  
23 [C#] public event EventHandler PreRequestHandlerExecute;  
24 [C++] public: \_\_event EventHandler\* PreRequestHandlerExecute;  
25 [VB] Public Event PreRequestHandlerExecute As EventHandler

1  
2 *Description*  
3  
4

5 [C#] public event EventHandler PreSendRequestContent;  
6 [C++] public: \_\_event EventHandler\* PreSendRequestContent;  
7 [VB] Public Event PreSendRequestContent As EventHandler  
8

9 *Description*  
10  
11

12 [C#] public event EventHandler PreSendRequestHeaders;  
13 [C++] public: \_\_event EventHandler\* PreSendRequestHeaders;  
14 [VB] Public Event PreSendRequestHeaders As EventHandler  
15

16 *Description*  
17  
18

19 [C#] public event EventHandler ReleaseRequestState;  
20 [C++] public: \_\_event EventHandler\* ReleaseRequestState;  
21 [VB] Public Event ReleaseRequestState As EventHandler  
22

23 *Description*  
24  
25

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#]      public      event      EventHandler      ResolveRequestCache;  
[C++]     public:     __event     EventHandler*      ResolveRequestCache;  
[VB]      Public      Event      ResolveRequestCache  As      EventHandler
```

*Description*

```
[C#].      public      event      EventHandler      UpdateRequestCache;  
[C++]     public:     __event     EventHandler*      UpdateRequestCache;  
[VB]      Public      Event      UpdateRequestCache  As      EventHandler
```

*Description*

Methods:  
AddOnAcquireRequestStateAsync

```
[C#] public void AddOnAcquireRequestStateAsync(BeginEventHandler bh,  
EndEventHandler eh);  
[C++] public: void AddOnAcquireRequestStateAsync(BeginEventHandler* bh,  
EndEventHandler* eh);  
[VB] Public Sub AddOnAcquireRequestStateAsync(ByVal bh As  
BeginEventHandler, ByVal eh As EndEventHandler)  
[JScript] public function AddOnAcquireRequestStateAsync(bh :  
BeginEventHandler, eh : EndEventHandler);
```

1  
2 *Description*

3  
4 AddOnAuthenticateRequestAsync

5  
6 [C#] public void AddOnAuthenticateRequestAsync(BeginEventHandler bh,  
7 EndEventHandler eh);

8 [C++] public: void AddOnAuthenticateRequestAsync(BeginEventHandler\* bh,  
9 EndEventHandler\* eh);

10 [VB] Public Sub AddOnAuthenticateRequestAsync(ByVal bh As  
11 BeginEventHandler, ByVal eh As EndEventHandler)

12 [JScript] public function AddOnAuthenticateRequestAsync(bh :  
13 BeginEventHandler, eh : EndEventHandler);

14  
15 *Description*

16  
17 AddOnAuthorizeRequestAsync

18  
19 [C#] public void AddOnAuthorizeRequestAsync(BeginEventHandler bh,  
20 EndEventHandler eh);

21 [C++] public: void AddOnAuthorizeRequestAsync(BeginEventHandler\* bh,  
22 EndEventHandler\* eh);

23 [VB] Public Sub AddOnAuthorizeRequestAsync(ByVal bh As  
24 BeginEventHandler, ByVal eh As EndEventHandler)

25 [JScript] public function AddOnAuthorizeRequestAsync(bh : BeginEventHandler,



1 eh : EndEventHandler);

2  
3 *Description*

4  
5 AddOnBeginRequestAsync

6  
7 [C#] public void AddOnBeginRequestAsync(BeginEventHandler bh,  
8 EndEventHandler eh);

9 [C++] public: void AddOnBeginRequestAsync(BeginEventHandler\* bh,  
10 EndEventHandler\* eh);

11 [VB] Public Sub AddOnBeginRequestAsync(ByVal bh As BeginEventHandler,  
12 ByVal eh As EndEventHandler)

13 [JScript] public function AddOnBeginRequestAsync(bh : BeginEventHandler, eh :  
14 EndEventHandler);

15  
16 *Description*

17  
18 AddOnEndRequestAsync

19  
20 [C#] public void AddOnEndRequestAsync(BeginEventHandler bh,  
21 EndEventHandler eh);

22 [C++] public: void AddOnEndRequestAsync(BeginEventHandler\* bh,  
23 EndEventHandler\* eh);

24 [VB] Public Sub AddOnEndRequestAsync(ByVal bh As BeginEventHandler,  
25 ByVal eh As EndEventHandler)

1 [JScript] public function AddOnEndRequestAsync(bh : BeginEventHandler, eh :  
2 EndEventHandler);

3  
4 *Description*

5  
6 AddOnPostRequestHandlerExecuteAsync

7  
8 [C#] public void AddOnPostRequestHandlerExecuteAsync(BeginEventHandler  
9 bh, EndEventHandler eh);

10 [C++] public: void  
11 AddOnPostRequestHandlerExecuteAsync(BeginEventHandler\* bh,  
12 EndEventHandler\* eh);

13 [VB] Public Sub AddOnPostRequestHandlerExecuteAsync(ByVal bh As  
14 BeginEventHandler, ByVal eh As EndEventHandler)

15 [JScript] public function AddOnPostRequestHandlerExecuteAsync(bh :  
16 BeginEventHandler, eh : EndEventHandler);

17  
18 *Description*

19  
20 AddOnPreRequestHandlerExecuteAsync

21  
22 [C#] public void AddOnPreRequestHandlerExecuteAsync(BeginEventHandler bh,  
23 EndEventHandler eh);

24 [C++] public: void AddOnPreRequestHandlerExecuteAsync(BeginEventHandler\*  
25 bh, EndEventHandler\* eh);

```

1 [VB] Public Sub AddOnPreRequestHandlerExecuteAsync(ByVal bh As
2 BeginEventHandler, ByVal eh As EndEventHandler)
3 [JScript] public function AddOnPreRequestHandlerExecuteAsync(bh :
4 BeginEventHandler, eh : EndEventHandler);

```

#### *Description*

#### AddOnReleaseRequestStateAsync

```

10 [C#] public void AddOnReleaseRequestStateAsync(BeginEventHandler bh,
11 EndEventHandler eh);
12 [C++] public: void AddOnReleaseRequestStateAsync(BeginEventHandler* bh,
13 EndEventHandler* eh);
14 [VB] Public Sub AddOnReleaseRequestStateAsync(ByVal bh As
15 BeginEventHandler, ByVal eh As EndEventHandler)
16 [JScript] public function AddOnReleaseRequestStateAsync(bh :
17 BeginEventHandler, eh : EndEventHandler);

```

#### *Description*

#### AddOnResolveRequestCacheAsync

```

23 [C#] public void AddOnResolveRequestCacheAsync(BeginEventHandler bh,
24 EndEventHandler eh);
25 [C++] public: void AddOnResolveRequestCacheAsync(BeginEventHandler* bh,

```

```

1 EndEventHandler*                                     eh);
2 [VB] Public Sub AddOnResolveRequestCacheAsync(ByVal bh As
3 BeginEventHandler, ByVal eh As EndEventHandler)
4 [JScript] public function AddOnResolveRequestCacheAsync(bh :
5 BeginEventHandler, eh : EndEventHandler);
6

```

### *Description*

```

9 AddOnUpdateRequestCacheAsync
10
11 [C#] public void AddOnUpdateRequestCacheAsync(BeginEventHandler bh,
12 EndEventHandler eh);
13 [C++] public: void AddOnUpdateRequestCacheAsync(BeginEventHandler* bh,
14 EndEventHandler* eh);
15 [VB] Public Sub AddOnUpdateRequestCacheAsync(ByVal bh As
16 BeginEventHandler, ByVal eh As EndEventHandler)
17 [JScript] public function AddOnUpdateRequestCacheAsync(bh :
18 BeginEventHandler, eh : EndEventHandler);
19

```

### *Description*

```

22 CompleteRequest
23
24 [C#] public void CompleteRequest();
25 [C++] public: void CompleteRequest();

```

1	[VB]	Public	Sub	CompleteRequest()
2	[JScript]	public	function	CompleteRequest();

3

4 *Description*

5

6 Dispose

7				
8	[C#]	public	virtual	void Dispose();
9	[C++]	public:	virtual	void Dispose();
10	[VB]	Overridable	Public	Sub Dispose()
11	[JScript]	public	function	Dispose();

12

13 *Description*

14 Cleans up the instance variables of an **HttpModule**.

15 The **System.Web.HttpApplication.Request** ,

16 **System.Web.HttpApplication.Response** ,

17 **System.Web.HttpApplication.Session** and

18 **System.Web.HttpApplication.Application** properties are not available for use at

19 the time **System.Web.HttpApplication.Dispose** is executed.

20 GetVaryByCustomString

21

22 [C#] public virtual string GetVaryByCustomString(HttpContext context, string

23 custom);

24 [C++] public: virtual String\* GetVaryByCustomString(HttpContext\* context,

25 String\* custom);

```

1 [VB] Overridable Public Function GetVaryByCustomString(ByVal context As
2 HttpContext, ByVal custom As String) As String
3 [JScript] public function GetVaryByCustomString(context : HttpContext, custom :
4 String) : String;

```

## *Description*

### Init

10	[C#]	public	virtual	void	Init();
11	[C++]	public:	virtual	void	Init();
12	[VB]	Overridable	Public	Sub	Init()
13	[JScript]	public	function		Init();

## *Description*

Initializes **HttpModule** instance variables and register event handlers with the hosting Application.

### IHttpAsyncHandler.BeginProcessRequest

```

20 [C#] IAsyncResult IHttpAsyncHandler.BeginProcessRequest(HttpContext
21 context, AsyncCallback cb, object ~extraData);
22 [C++] IAsyncResult* IHttpAsyncHandler::BeginProcessRequest(HttpContext*
23 context, AsyncCallback* cb, Object* extraData);
24 [VB] Function BeginProcessRequest(ByVal context As HttpContext, ByVal cb As
25 AsyncCallback, ByVal extraData As Object) As IAsyncResult Implements

```

1 IHttpAsyncHandler.BeginProcessRequest

2 [JScript] function IHttpAsyncHandler.BeginProcessRequest(context :  
3 HttpContext, cb : AsyncCallback, extraData : Object) : IAsyncResult;

4 IHttpAsyncHandler.EndProcessRequest

6 [C#] void IHttpAsyncHandler.EndProcessRequest(IAsyncResult result);

7 [C++] void IHttpAsyncHandler::EndProcessRequest(IAsyncResult\* result);

8 [VB] Sub EndProcessRequest(ByVal result As IAsyncResult) Implements  
9 IHttpAsyncHandler.EndProcessRequest

10 [JScript] function IHttpAsyncHandler.EndProcessRequest(result : IAsyncResult);

11 IHttpHandler.ProcessRequest

13 [C#] void IHttpHandler.ProcessRequest(HttpContext context);

14 [C++] void IHttpHandler::ProcessRequest(HttpContext\* context);

15 [VB] Sub ProcessRequest(ByVal context As HttpContext) Implements  
16 IHttpHandler.ProcessRequest

17 [JScript] function IHttpHandler.ProcessRequest(context : HttpContext);

18 HttpApplicationState class (System.Web)

19 ToString

## 22 *Description*

23 Enables sharing of global information across multiple sessions and requests  
24 within an ASP.NET application.

1 An ASP.NET application is the sum of all files, pages, handlers, modules,  
2 and code within the scope of a virtual directory and its subdirectories on a single  
3 web server.

4 AllKeys

5 ToString

6  
7 [C#] public string[] AllKeys {get;}

8 [C++] public: \_\_property String\* get\_AllKeys();

9 [VB] Public ReadOnly Property AllKeys As String ()

10 [JScript] public function get AllKeys() : String[];

11  
12 *Description*

13 Gets the access keys in the **System.Web.HttpApplicationState** collection.

14 Contents

15 ToString

16  
17 [C#] public HttpApplicationState Contents {get;}

18 [C++] public: \_\_property HttpApplicationState\* get\_Contents();

19 [VB] Public ReadOnly Property Contents As HttpApplicationState

20 [JScript] public function get Contents() : HttpApplicationState;

21  
22 *Description*

23 Gets a reference to the **System.Web.HttpApplicationState** object.

24 This property provides compatibility with earlier versions of ASP.

25 Count



ToString

```
[C#]      public      override      int      Count      {get;}
[C++]     public:     __property     virtual    int      get_Count();
[VB]      Overrides   Public    ReadOnly   Property   Count    As    Integer
[JScript] public      function    get      Count()    :      int;
```

*Description*

Gets the number of objects in the **System.Web.HttpApplicationState** collection.

IsReadOnly

Item

ToString

**System.Web.HttpApplicationState**

*Description*

Gets the value of a single **System.Web.HttpApplicationState** object by name. The name of the object in the collection.

Item

ToString

```
[C#]      public      object      this[int      index]      {get;}
[C++]     public:     __property     Object*      get_Item(int      index);
[VB]      Public Default ReadOnly Property Item(ByVal index As Integer) As Object
[JScript] returnValue =      HttpApplicationStateObject.Item(index);
```

1  
2 *Description*

3 Gets a single **System.Web.HttpApplicationState** object by index. The  
4 numerical index of the object in the collection.

5 Keys

6 StaticObjects

7 ToString

8  
9  
10 *Description*

11 Gets all objects declared via an tag within the ASP.NET application.

12 Application objects are defined in the Global.asax file.

13 Add

14  
15 [C#] public void Add(string name, object value);

16 [C++] public: void Add(String\* name, Object\* value);

17 [VB] Public Sub Add(ByVal name As String, ByVal value As Object)

18 [JScript] public function Add(name : String, value : Object);

19  
20 *Description*

21 Adds a new object to the **System.Web.HttpApplicationState** collection.

22 The name of the object to be added to the collection. The value of the object.

23 Clear

24  
25 [C#] public void Clear();

1	[C++]	public:	void	Clear();
2	[VB]	Public	Sub	Clear()
3	[JScript]	public	function	Clear();

4

5 *Description*

6 Removes all objects from an **System.Web.HttpApplicationState**

7 collection.

8 Get

9

10	[C#]	public	object	Get(int	index);
11	[C++]	public:	Object*	Get(int	index);
12	[VB]	Public	Function	Get(ByVal index As Integer) As Object	
13	[JScript]	public	function	Get(index : int) : Object;	

14

15 *Description*

16 Gets an **System.Web.HttpApplicationState** object by numerical index.

17 *Return Value:* The object referenced by *index* . The index of the application state

18 object.

19 Get

20

21	[C#]	public	object	Get(string	name);
22	[C++]	public:	Object*	Get(String*	name);
23	[VB]	Public	Function	Get(ByVal name As String) As Object	
24	[JScript]	public	function	Get(name : String) : Object;	Gets an
25	<b>System.Web.HttpApplicationState</b> object by name or index.				

1  
2 *Description*

3 Gets an **System.Web.HttpApplicationState** object by name.

4 *Return Value:* The object referenced by *name*.

5 The following example returns an object named MyAppVar1 from the  
6 **System.Web.HttpApplicationState** collection of the intrinsic  
7 **System.Web.HttpContext.Application** object and copies it to a new object  
8 variable. The name of the object.

9 **GetKey**

10  
11 [C#] public string GetKey(int index);

12 [C++] public: String\* GetKey(int index);

13 [VB] Public Function GetKey(ByVal index As Integer) As String

14 [JScript] public function GetKey(index : int) : String;

15  
16 *Description*

17 Gets an **System.Web.HttpApplicationState** object name by index.

18 *Return Value:* The name under which the application state object was saved. The  
19 index of the application state object.

20 **Lock**

21  
22 [C#] public void Lock();

23 [C++] public: void Lock();

24 [VB] Public Sub Lock()

25 [JScript] public function Lock();

1  
2 *Description*

3 Locks access to an **System.Web.HttpApplicationState** variable to  
4 facilitate access synchronization.

5 Remove

6  
7 [C#] public void Remove(string name);

8 [C++] public: void Remove(String\* name);

9 [VB] Public Sub Remove(ByVal name As String)

10 [JScript] public function Remove(name : String);

11  
12 *Description*

13 Removes the named object from an **System.Web.HttpApplicationState**  
14 collection. The name of the object to be removed from the collection.

15 RemoveAll

16  
17 [C#] public void RemoveAll();

18 [C++] public: void RemoveAll();

19 [VB] Public Sub RemoveAll()

20 [JScript] public function RemoveAll();

21  
22 *Description*

23 Removes all objects from an **System.Web.HttpApplicationState**  
24 collection.

**System.Web.HttpApplicationState.RemoveAll** is an internal call to **System.Web.HttpApplicationState.Clear**.

### RemoveAt

[C#]            public            void            RemoveAt(int            index);

[C++]           public:            void            RemoveAt(int            index);

[VB]    Public    Sub    RemoveAt(ByVal    index    As    Integer)

[JScript] public function RemoveAt(index : int); Removes an object from the application state collection by name.

### Set

[C#]    public    void    Set(string    name,    object    value);

[C++]    public:    void    Set(String\*    name,    Object\*    value);

[VB]    Public    Sub    Set(ByVal    name    As    String,    ByVal    value    As    Object)

[JScript]    public    function    Set(name    :    String,    value    :    Object);

### *Description*

Updates the value of an object in an **System.Web.HttpApplicationState** collection. The name of the object to be updated. The updated value of the object.

### UnLock

[C#]            public            void            UnLock();

[C++]           public:            void            UnLock();

[VB]            Public            Sub            UnLock()

[JScript]           public            function            UnLock();

1  
2 *Description*

3 Unlocks access to an **System.Web.HttpApplicationState** variable to  
4 facilitate access synchronization.

5 **HttpBrowserCapabilities** class (System.Web)

6 **UnLock**

7  
8  
9 *Description*

10 Enables the server to gather information on the capabilities of the browser  
11 that is running on the client.

12 **System.Web.HttpBrowserCapabilities** properties are accessible through  
13 the **System.Web.HttpRequest.Browser** property of ASP.NET's intrinsic  
14 **System.Web.HttpContext.Request** object.

15 **HttpBrowserCapabilities**

16 *Example Syntax:*

17 **UnLock**

18  
19 [C#] public **HttpBrowserCapabilities();**

20 [C++] public: **HttpBrowserCapabilities();**

21 [VB] Public Sub **New()**

22 [JScript] public function **HttpBrowserCapabilities();**

23 **ActiveXControls**

24 **UnLock**

```

1
2 [C#]          public          bool          ActiveXControls          {get;}
3 [C++]         public:         __property    bool          get_ActiveXControls();
4 [VB]   Public  ReadOnly  Property  ActiveXControls  As  Boolean
5 [JScript]  public  function  get  ActiveXControls()  :  Boolean;
6

```

#### 7 *Description*

8 Gets a value indicating whether the client browser supports ActiveX  
9 controls.

10 AOL

11 UnLock

```

12
13 [C#]          public          bool          AOL          {get;}
14 [C++]         public:         __property    bool          get_AOL();
15 [VB]   Public  ReadOnly  Property  AOL  As  Boolean
16 [JScript]  public  function  get  AOL()  :  Boolean;
17

```

#### 18 *Description*

19 Gets a value indicating whether the client is an America Online (AOL)  
20 browser.

21 BackgroundSounds

22 UnLock

```

23
24 [C#]          public          bool          BackgroundSounds          {get;}
25 [C++]         public:         __property    bool          get_BackgroundSounds();

```



```

1  [VB]   Public   ReadOnly   Property   BackgroundSounds   As   Boolean
2  [JScript]   public   function   get   BackgroundSounds()   :   Boolean;

```

3

4 *Description*

5 Gets a value indicating whether the client browser supports background

6 sounds.

7 Beta

8 UnLock

9

10 [C#] public bool Beta {get;}

11 [C++] public: \_\_property bool get\_Beta();

12 [VB] Public ReadOnly Property Beta As Boolean

13 [JScript] public function get Beta() : Boolean;

14

15 *Description*

16 Gets a value indicating whether the browser is a beta release.

17 Browser

18 UnLock

19

20 [C#] public string Browser {get;}

21 [C++] public: \_\_property String\* get\_Browser();

22 [VB] Public ReadOnly Property Browser As String

23 [JScript] public function get Browser() : String;

24

25 *Description*

Gets the browser string (if any) that was transmitted in the **User-Agent** header.

CDF

UnLock

[C#]                    public                    bool                    CDF                    {get;}

[C++]                  public:                  \_\_property                  bool                  get\_CDF();

[VB]    Public    ReadOnly    Property    CDF    As    Boolean

[JScript]    public    function    get    CDF()    :    Boolean;

#### *Description*

Gets a value indicating whether the client browser supports Channel Definition Format (CDF) for webcasting.

ClrVersion

UnLock

[C#]                    public                    Version                    ClrVersion                    {get;}

[C++]                  public:                  \_\_property                  Version\*                  get\_ClrVersion();

[VB]    Public    ReadOnly    Property    ClrVersion    As    Version

[JScript]    public    function    get    ClrVersion()    :    Version;

#### *Description*

Gets the version number of the .NET common language runtime that the client browser supports.

If no common language runtime version is specified, the property value is 0, 0,-1,-1.

Cookies

UnLock

[C#] public bool Cookies {get;}

[C++] public: \_\_property bool get\_Cookies();

[VB] Public ReadOnly Property Cookies As Boolean

[JScript] public function get Cookies() : Boolean;

#### Description

Gets a value indicating whether the client browser supports cookies.

Crawler

UnLock

[C#] public bool Crawler {get;}

[C++] public: \_\_property bool get\_Crawler();

[VB] Public ReadOnly Property Crawler As Boolean

[JScript] public function get Crawler() : Boolean;

#### Description

Gets a value indicating whether the client browser is a Web crawler search engine.

EcmaScriptVersion

UnLock

```

1
2 [C#]      public      Version      EcmaScriptVersion      {get;}
3 [C++]     public:     __property   Version*      get_EcmaScriptVersion();
4 [VB]      Public     ReadOnly     Property   EcmaScriptVersion   As   Version
5 [JScript] public     function   get     EcmaScriptVersion()   :   Version;
6

```

### 7 *Description*

8 Gets the version number of ECMA script that the client browser supports.

9 The European Computer Manufacturer's Association develops standards for  
10 information and communication systems. For more information, see ECMA's  
11 official Web site at <http://www.ecma.ch>.

12 Frames

13 UnLock

```

14
15 [C#]      public      bool      Frames      {get;}
16 [C++]     public:     __property   bool      get_Frames();
17 [VB]      Public     ReadOnly     Property   Frames      As   Boolean
18 [JScript] public     function   get     Frames()      :   Boolean;
19

```

### 20 *Description*

21 Gets a value indicating whether the client browser supports HTML frames.

22 Item

23 JavaApplets

24 UnLock

1  
2  
3 *Description*

4 Gets a value indicating whether the client browser supports Java applets.

5 JavaScript

6 UnLock

7  
8 [C#] public bool JavaScript {get;}

9 [C++] public: \_\_property bool get\_JavaScript();

10 [VB] Public ReadOnly Property JavaScript As Boolean

11 [JScript] public function get JavaScript() : Boolean;

12  
13 *Description*

14 Gets a value indicating whether the client browser supports JavaScript.

15 MajorVersion

16 UnLock

17  
18 [C#] public int MajorVersion {get;}

19 [C++] public: \_\_property int get\_MajorVersion();

20 [VB] Public ReadOnly Property MajorVersion As Integer

21 [JScript] public function get MajorVersion() : int;

22  
23 *Description*

24 Gets the major (that is, integer) version number of the client browser.

25 MinorVersion

## UnLock

```
[C#]      public      double      MinorVersion      {get;}
[C++]     public:     __property  double      get_MinorVersion();
[VB]      Public      ReadOnly    Property  MinorVersion  As      Double
[JScript] public      function  get      MinorVersion()  :      double;
```

### *Description*

Gets the minor (that is, decimal) version number of the client browser.

MSDomVersion

UnLock

```
[C#]      public      Version      MSDomVersion      {get;}
[C++]     public:     __property  Version*      get_MSDomVersion();
[VB]      Public      ReadOnly    Property  MSDomVersion  As      Version
[JScript] public      function  get      MSDomVersion()  :      Version;
```

### *Description*

Gets the version of Microsoft HTML (MSHTML) Document Object Model (DOM) that the client browser supports.

Platform

UnLock

```
[C#]      public      string      Platform      {get;}
[C++]     public:     __property  String*      get_Platform();
```

1	[VB]	Public	ReadOnly	Property	Platform	As	String
2	[JScript]	public	function	get	Platform()	:	String;

3

4 *Description*

5 Gets the name of the platform that the client uses.

6 Some possible **Platform** values are: Unknown, Win95, Win98, WinNT

7 (which includes Windows 2000), Win16, WinCE, Mac68K, MacPPC, UNIX, and

8 WebTV.

9 Tables

10 UnLock

12	[C#]	public	bool	Tables	{get;}
13	[C++]	public:	__property	bool	get_Tables();
14	[VB]	Public	ReadOnly	Property	Tables As Boolean
15	[JScript]	public	function	get	Tables() : Boolean;

16

17 *Description*

18 Gets a value indicating whether the client browser supports HTML tables.

19 TagWriter

20 UnLock

22	[C#]	public	Type	TagWriter	{get;}
23	[C++]	public:	__property	Type*	get_TagWriter();
24	[VB]	Public	ReadOnly	Property	TagWriter As Type
25	[JScript]	public	function	get	TagWriter() : Type;

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

*Description*

Type  
UnLock

[C#]	public	string	Type	{get;}
[C++]	public:	__property	String*	get_Type();
[VB]	Public	ReadOnly	Property	Type As String
[JScript]	public	function	get	Type() : String;

*Description*

Gets the name and major (that is, integer) version number of the client browser.

VBScript  
UnLock

[C#]	public	bool	VBScript	{get;}
[C++]	public:	__property	bool	get_VBScript();
[VB]	Public	ReadOnly	Property	VBScript As Boolean
[JScript]	public	function	get	VBScript() : Boolean;

*Description*

Gets a value indicating whether the client browser supports VBScript.  
Version



```

1      UnLock
2
3  [C#]      public      string      Version      {get;}
4  [C++]     public:     __property  String*      get_Version();
5  [VB]      Public     ReadOnly    Property    Version    As      String
6  [JScript] public     function    get      Version()    :      String;
7

```

#### *Description*

Gets the full (integer and decimal) version number of the client browser.

W3CDomVersion

UnLock

```

13 [C#]      public      Version      W3CDomVersion      {get;}
14 [C++]     public:     __property  Version*      get_W3CDomVersion();
15 [VB]      Public     ReadOnly    Property    W3CDomVersion    As      Version
16 [JScript] public     function    get      W3CDomVersion()    :      Version;
17

```

#### *Description*

Gets the version of the World Wide Web Consortium (W3C) XML Document Object Model (DOM) that the client browser supports.

Win16

UnLock

```

24 [C#]      public      bool      Win16      {get;}
25 [C++]     public:     __property  bool      get_Win16();

```

```

1 [VB]      Public      ReadOnly      Property      Win16      As      Boolean
2 [JScript]      public      function      get      Win16()      :      Boolean;

```

#### *Description*

Gets a value indicating whether the client is a Win16-based machine.

Win32

UnLock

```

9 [C#]      public      bool      Win32      {get;}

```

```

10 [C++]      public:      __property      bool      get_Win32();

```

```

11 [VB]      Public      ReadOnly      Property      Win32      As      Boolean

```

```

12 [JScript]      public      function      get      Win32()      :      Boolean;

```

#### *Description*

Gets a value indicating whether the client is a Win32-based machine.

HttpCacheability enumeration (System.Web)

ToString

#### *Description*

Provides enumerated values that are used to set the **Cache-Control** HTTP header.

ToString

```

25 [C#]      public      const      HttpCacheability      NoCache;

```

[C++]	public:	const	HttpCacheability	NoCache;
[VB]	Public	Const	NoCache	As HttpCacheability
[JScript]	public	var	NoCache	: HttpCacheability;

#### Description

Sets the **Cache-Control: no-cache** header. Without a field name, the directive applies to the entire request and a shared (proxy server) cache must force a successful revalidation with the origin Web server before satisfying the request. With a field name, the directive applies only to the named field; the rest of the response may be supplied from a shared cache.

#### ToString

[C#]	public	const	HttpCacheability	Private;
[C++]	public:	const	HttpCacheability	Private;
[VB]	Public	Const	Private	As HttpCacheability
[JScript]	public	var	Private	: HttpCacheability;

#### Description

Default value. Sets **Cache-Control: private** to specify that the response is cacheable only on the client and not by shared (proxy server) caches.

#### ToString

[C#]	public	const	HttpCacheability	Public;
[C++]	public:	const	HttpCacheability	Public;
[VB]	Public	Const	Public	As HttpCacheability

1 [JScript] public var Public : HttpCacheability;

2

3 *Description*

4 Sets **Cache-Control: public** to specify that the response is cacheable by  
5 clients and shared (proxy) caches.

6 ToString

7

8 [C#] public const HttpCacheability Server;

9 [C++] public: const HttpCacheability Server;

10 [VB] Public Const Server As HttpCacheability

11 [JScript] public var Server : HttpCacheability;

12

13 *Description*

14 Specifies that the response is cached only at the origin server. Similar to the  
15 **NoCache** option. Clients receive a **Cache-Control: no-cache** directive but the  
16 document is cached on the origin server.

17 HttpCachePolicy class (System.Web)

18 ToString

19

20

21 *Description*

22 Contains methods for setting cache-specific HTTP headers and for  
23 controlling the ASP.NET page output cache.

24

25

For background information on HTTP headers and controlling caching, see the document RFC 2616: Hypertext Transfer Protocol - HTTP/1.1, available on the World Wide Web Consortium's site at <http://www.w3c.org>.

VaryByHeaders

ToString

```
[C#]    public    HttpCacheVaryByHeaders    VaryByHeaders    {get;}
```

```
[C++] public: __property HttpCacheVaryByHeaders* get_VaryByHeaders();
```

```
[VB] Public ReadOnly Property VaryByHeaders As HttpCacheVaryByHeaders
```

```
[JScript] public function get VaryByHeaders() : HttpCacheVaryByHeaders;
```

#### *Description*

Gets the list of all HTTP headers that will be used to vary cache output.

When a cached item has several vary headers, a separate version of the requested document is available from the cache for each HTTP header type.

VaryByParams

ToString

```
[C#]    public    HttpCacheVaryByParams    VaryByParams    {get;}
```

```
[C++] public: __property HttpCacheVaryByParams* get_VaryByParams();
```

```
[VB] Public ReadOnly Property VaryByParams As HttpCacheVaryByParams
```

```
[JScript] public function get VaryByParams() : HttpCacheVaryByParams;
```

#### *Description*

1 Gets the list of parameters received by a **GET** (querystring) or **POST** (in  
2 the body of the HTTP request) that affect caching.

3 For each named parameter in **VaryByParams** a separate version of the  
4 requested document is available from the cache, the version varying by the  
5 parameter's value.

#### 6 AddValidationCallback

7  
8 [C#] public void AddValidationCallback(HttpCacheValidateHandler handler,  
9 object data);

10 [C++] public: void AddValidationCallback(HttpCacheValidateHandler\* handler,  
11 Object\* data);

12 [VB] Public Sub AddValidationCallback(ByVal handler As  
13 HttpCacheValidateHandler, ByVal data As Object)

14 [JScript] public function AddValidationCallback(handler :  
15 HttpCacheValidateHandler, data : Object);

#### 16 17 *Description*

18 Registers a validation callback for the current response.

19 **AddValidationCallback** provides a mechanism to programmatically check  
20 the validity of a item in the cache before the item is returned from the cache. The  
21 **System.Web.HttpCacheValidateHandler** value. The arbitrary user-supplied data  
22 that is passed back to the **AddValidationCallback** delegate.

#### 23 AppendCacheExtension

24  
25 [C#] public void AppendCacheExtension(string extension);

```

1 [C++] public: void AppendCacheExtension(String* extension);
2 [VB] Public Sub AppendCacheExtension(ByVal extension As String)
3 [JScript] public function AppendCacheExtension(extension : String);

```

#### *Description*

Appends the specified text to the **Cache-Control** HTTP header.

If the browser does not recognize cache control directives or extensions, the browser must ignore the unrecognized terms. For more information, see the document RFC 2616: Hypertext Transfer Protocol - HTTP/1.1, available on the World Wide Web Consortium's site at <http://www.w3c.org> . The text to append to the **Cache-Control** header.

#### *SetCacheability*

```

14 [C#] public void SetCacheability(HttpCacheability cacheability);
15 [C++] public: void SetCacheability(HttpCacheability cacheability);
16 [VB] Public Sub SetCacheability(ByVal cacheability As HttpCacheability)
17 [JScript] public function SetCacheability(cacheability : HttpCacheability); Sets the
18 Cache-Control HTTP header. The Cache-Control HTTP header controls how
19 documents are to be cached on the network.

```

#### *Description*

Sets the **Cache-Control** header to one of the values of **System.Web.HttpCacheability** . An **System.Web.HttpCacheability** enumeration value.

#### *SetCacheability*

```

1
2 [C#] public void SetCacheability(HttpCacheability cacheability, string field);
3 [C++] public: void SetCacheability(HttpCacheability cacheability, String* field);
4 [VB] Public Sub SetCacheability(ByVal cacheability As HttpCacheability, ByVal
5 field                                     As                                     String)
6 [JScript] public function SetCacheability(cacheability : HttpCacheability, field :
7 String);
8

```

### *Description*

Sets the **Cache-Control** header to one of the values of **System.Web.HttpCacheability** and appends an extension to the directive.

The field name extension is valid only when used with the **private** or **no-cache** directives. For more information, see the document RFC 2616: Hypertext Transfer Protocol - HTTP/1.1, available on the World Wide Web Consortium's site at <http://www.w3c.org>. The **System.Web.HttpCacheability** enumeration value to set the header to. The cache control extension to add to the header.

### **SetETag**

```

17
18
19 [C#]          public          void          SetETag(string          etag);
20 [C++]          public:          void          SetETag(String*          etag);
21 [VB]    Public    Sub    SetETag(ByVal    etag    As    String)
22 [JScript]    public    function    SetETag(etag    :    String);
23

```

### *Description*

Sets the **ETag** HTTP header to the specified string.



The ETag header is a unique identifier for a specific version of a document. Once an **ETag** header is set, subsequent attempts to set it will fail and an exception will be thrown. The text to use for the **ETag** header.

#### SetETagFromFileDependencies

[C#]	public	void	SetETagFromFileDependencies();
[C++]	public:	void	SetETagFromFileDependencies();
[VB]	Public	Sub	SetETagFromFileDependencies()
[JScript]	public	function	SetETagFromFileDependencies();

#### *Description*

Sets the **ETag** HTTP header based on the time stamps of the handler's file dependencies.

**SetETagFromFileDependencies** sets the **ETag** header by retrieving the last modified time stamps of all files on which the handler is dependent, combining all file names and time stamps into a single string, then hashing that string into a single digest that is used as the **ETag**.

#### SetExpires

[C#]	public	void	SetExpires(DateTime	date);
[C++]	public:	void	SetExpires(DateTime	date);
[VB]	Public	Sub	SetExpires(ByVal	date As DateTime)
[JScript]	public	function	SetExpires(date	: DateTime);

#### *Description*

1 Sets the **Expires** HTTP header to an absolute date and time.

2 This method will fail if the expiration date violates the principle of  
3 restrictiveness. The absolute **System.DateTime** value to set the **Expires** header to.

4 **SetLastModified**

5  
6 [C#] public void SetLastModified(DateTime date);

7 [C++] public: void SetLastModified(DateTime date);

8 [VB] Public Sub SetLastModified(ByVal date As DateTime)

9 [JScript] public function SetLastModified(date : DateTime);

10  
11 *Description*

12 Sets the **Last-Modified** HTTP header to the **System.DateTime** value  
13 supplied.

14 The **Last-Modified** HTTP header time stamps the document with the  
15 **DateTime** value indicating when the document was last modified. The new  
16 **System.DateTime** value for the **Last-Modified** header.

17 **SetLastModifiedFromFileDependencies**

18  
19 [C#] public void SetLastModifiedFromFileDependencies();

20 [C++] public: void SetLastModifiedFromFileDependencies();

21 [VB] Public Sub SetLastModifiedFromFileDependencies()

22 [JScript] public function SetLastModifiedFromFileDependencies();

23  
24 *Description*

1       Sets the **Last-Modified** HTTP header based on the time stamps of the  
2 handler's file dependencies.

### 3       SetMaxAge

4  
5   [C#]       public       void       SetMaxAge(TimeSpan       delta);  
6   [C++]       public:       void       SetMaxAge(TimeSpan       delta);  
7   [VB]       Public       Sub       SetMaxAge(ByVal       delta       As       TimeSpan)  
8   [JScript]   public       function   SetMaxAge(delta       :       TimeSpan);  
9

### 10      *Description*

11       Sets the **Cache-Control: max-age** HTTP header based on the specified  
12 time span.

13       **Max-age** is the maximum absolute time a document is allowed to exist  
14 before being considered stale. The time span used to set the **Cache-Control: max-**  
15 **age** header.

### 16       SetNoServerCaching

17  
18   [C#]       public       void       SetNoServerCaching();  
19   [C++]       public:       void       SetNoServerCaching();  
20   [VB]       Public       Sub       SetNoServerCaching()  
21   [JScript]   public       function   SetNoServerCaching();  
22

### 23      *Description*

24       Stops all origin-server caching for the current response.  
25

Explicitly denies caching of the document on the origin-server. Once set, all requests for the document are fully processed. When this method is invoked, caching cannot be reenabled for the current response.

#### SetNoStore

[C#]	public	void	SetNoStore();
[C++]	public:	void	SetNoStore();
[VB]	Public	Sub	SetNoStore()
[JScript]	public	function	SetNoStore();

#### *Description*

Sets the **Cache-Control: no-store** directive.

#### SetNoTransforms

[C#]	public	void	SetNoTransforms();
[C++]	public:	void	SetNoTransforms();
[VB]	Public	Sub	SetNoTransforms()
[JScript]	public	function	SetNoTransforms();

#### *Description*

Sets the **CacheControl: no-transform** directive.

The **no-transform CacheControl** setting instructs network caching applications to not modify the document.

#### SetProxyMaxAge

```

1
2 [C#]      public      void      SetProxyMaxAge(TimeSpan      delta);
3 [C++]     public:     void      SetProxyMaxAge(TimeSpan      delta);
4 [VB]      Public     Sub      SetProxyMaxAge(ByVal      delta      As      TimeSpan)
5 [JScript] public      function SetProxyMaxAge(delta      :      TimeSpan);
6

```

### *Description*

Sets the **Cache-Control: s-maxage** HTTP header based on the specified time span.

**System.Web.HttpCachePolicy.SetProxyMaxAge(System.TimeSpan)** does not use sliding expiration and will fail if the expiration date violates the principle of restrictiveness. The time span used to set the **Cache-Control: s-maxage** header.

### **SetRevalidation**

```

14
15
16 [C#]      public      void      SetRevalidation(HttpCacheRevalidation      revalidation);
17 [C++]     public:     void      SetRevalidation(HttpCacheRevalidation      revalidation);
18 [VB]      Public     Sub      SetRevalidation(ByVal      revalidation      As      HttpCacheRevalidation)
19 [JScript] public      function SetRevalidation(revalidation : HttpCacheRevalidation);
20

```

### *Description*

Sets the **Cache-Control** HTTP header to either the **must-revalidate** or the **proxy-revalidate** directives based on the supplied enumeration value.

The default is to send neither directive in a header unless explicitly specified by this method. The **System.Web.HttpCacheRevalidation** enumeration value to set the **Cache-Control** header to.

#### SetSlidingExpiration

[C#]        public        void        SetSlidingExpiration(bool        slide);

[C++]        public:        void        SetSlidingExpiration(bool        slide);

[VB]    Public    Sub    SetSlidingExpiration(ByVal    slide    As    Boolean)

[JScript]    public    function    SetSlidingExpiration(slide    :    Boolean);

#### *Description*

Sets cache expiration to sliding.

When cache expiration is set to sliding, the **Cache-Control** HTTP header will be renewed with each response. This expiration mode is identical to the IIS configuration option to add an expiration header to all output set relative to the current time. **true** or **false** .

#### SetValidUntilExpires

[C#]        public        void        SetValidUntilExpires(bool        validUntilExpires);

[C++]        public:        void        SetValidUntilExpires(bool        validUntilExpires);

[VB]    Public    Sub    SetValidUntilExpires(ByVal    validUntilExpires    As    Boolean)

[JScript]    public    function    SetValidUntilExpires(validUntilExpires    :    Boolean);

#### *Description*

## SetVaryByCustom

```
[C#]      public      void      SetVaryByCustom(string      custom);
[C++]     public:     void      SetVaryByCustom(String*      custom);
[VB]      Public      Sub      SetVaryByCustom(ByVal      custom      As      String)
[JScript] public      function SetVaryByCustom(custom      :      String);
```

### *Description*

Sets the **Vary** HTTP header to the specified text string. The text to set the **Vary** header to.

HttpCacheRevalidation enumeration (System.Web)

ToString

### *Description*

Provides enumerated values that are used to set revalidation-specific **Cache-Control** HTTP headers.

ToString

```
[C#]      public      const      HttpCacheRevalidation      AllCaches;
[C++]     public:     const      HttpCacheRevalidation      AllCaches;
[VB]      Public      Const      AllCaches      As      HttpCacheRevalidation
[JScript] public      var      AllCaches      :      HttpCacheRevalidation;
```

### *Description*

1       Sets the **Cache-Control: must-revalidate** HTTP header.

2       ToString

3

4   [C#]       public       const       HttpCacheRevalidation       None;

5   [C++]       public:       const       HttpCacheRevalidation       None;

6   [VB]       Public       Const       None       As       HttpCacheRevalidation

7   [JScript]       public       var       None       :       HttpCacheRevalidation;

8

9   *Description*

10       Default value. If this value is set, no cache-revalidation directive is sent.

11       ToString

12

13   [C#]       public       const       HttpCacheRevalidation       ProxyCaches;

14   [C++]       public:       const       HttpCacheRevalidation       ProxyCaches;

15   [VB]       Public       Const       ProxyCaches       As       HttpCacheRevalidation

16   [JScript]       public       var       ProxyCaches       :       HttpCacheRevalidation;

17

18   *Description*

19       Sets the **Cache-Control: proxy-revalidate** HTTP header.

20       HttpCacheValidateHandler delegate (System.Web)

21       ToString

22

23

24   *Description*

25



1 Delegate method that is called when a cached item is validated. Cache  
2 items invalidated within the method are treated as cache misses. The  
3 **System.Web.HttpContext** object containing information about the current  
4 request. User-supplied data used to validate the cached item. A  
5 **System.Web.HttpValidationStatus** enumeration value.

6 If any handler invalidates the cached item, the item is evicted from the  
7 cache and the request is handled as a cache miss.

8 HttpCacheVaryByHeaders class (System.Web)

9 ToString

10  
11  
12 *Description*

13 Provides a type-safe way to set the **Vary** HTTP header.

14 The **Vary** header indicates the request-header fields that the server uses to  
15 determine which of multiple cached responses is sent in response to a client  
16 request.

17 AcceptTypes

18 ToString

19  
20 [C#] public bool AcceptTypes {get; set;}

21 [C++] public: \_\_property bool get\_AcceptTypes();public: \_\_property void  
22 set\_AcceptTypes(bool);

23 [VB] Public Property AcceptTypes As Boolean

24 [JScript] public function get AcceptTypes() : Boolean;public function set  
25 AcceptTypes(Boolean);

## Description

Gets or sets a value indicating whether the origin server adds the **Accept** field to the **Vary** HTTP header.

The **Vary** header indicates the request-header fields that the server uses to determine which of multiple cached responses is sent in response to a client request. The **Accept** field specifies that the server selects the response based on the media types acceptable to the client.

Item

ToString

```
[C#] public bool this[string header] {get; set;}
```

```
[C++] public: __property bool get_Item(String* header);public: __property void  
set_Item(String* header, bool);
```

```
[VB] Public Default Property Item(ByVal header As String) As Boolean
```

```
[JScript] returnValue =
```

```
HttpCacheVaryByHeadersObject.Item(header);HttpCacheVaryByHeadersObject.I  
tem(header) = returnValue;
```

## Description

Gets or sets a value indicating whether the origin server should add a custom field to the **Vary** HTTP header.

The **Vary** header indicates the request-header fields that the server uses to determine which of multiple cached responses is sent in response to a client request. The name of the custom header.

UserAgent

ToString

[C#]            public            bool            userAgent            {get;            set;}

[C++]   public:   \_\_property   bool   get\_UserAgent();public:   \_\_property   void  
set\_UserAgent(bool);

[VB]            Public            Property            userAgent            As            Boolean

[JScript]   public   function   get   userAgent() : Boolean;public   function   set  
UserAgent(Boolean);

### *Description*

Gets or sets a value indicating whether the origin server adds the **User-Agent** field to the **Vary** HTTP header.

The **Vary** header indicates the request-header fields that the server uses to determine which of multiple cached responses is sent in response to a client request. The **User-Agent** field specifies that the server selects the response based on the type of client user-agent.

UserCharSet

ToString

[C#]            public            bool            UserCharSet            {get;            set;}

[C++]   public:   \_\_property   bool   get\_UserCharSet();public:   \_\_property   void  
set\_UserCharSet(bool);

[VB]            Public            Property            UserCharSet            As            Boolean

[JScript]   public   function   get   UserCharSet() : Boolean;public   function   set

1 UserCharSet(Boolean);

3 *Description*

4 Gets or sets a value indicating whether the origin server should add the  
5 **Accept-Charset** field to the **Vary** HTTP header.

6 The **Vary** header indicates the request-header fields that the server uses to  
7 determine which of multiple cached responses is sent in response to a client  
8 request. The **Accept-CharSet** field specifies that the server selects the response  
9 based on the client's character set.

10 UserLanguage

11 ToString

12  
13 [C#] public bool UserLanguage {get; set;}

14 [C++] public: \_\_property bool get\_UserLanguage();public: \_\_property void  
15 set\_UserLanguage(bool);

16 [VB] Public Property UserLanguage As Boolean

17 [JScript] public function get UserLanguage() : Boolean;public function set  
18 UserLanguage(Boolean);

19  
20 *Description*

21 Gets or sets a value indicating whether the origin server adds the **Accept-**  
22 **Language** field to the **Vary** HTTP header.

23 The **Vary** header indicates the request-header fields that the server uses to  
24 determine which of multiple cached responses is sent in response to a client  
25

request. The **Accept-Language** field specifies that the server selects the response based on languages acceptable to the client.

### VaryByUnspecifiedParameters

[C#]	public	void	VaryByUnspecifiedParameters();
[C++]	public:	void	VaryByUnspecifiedParameters();
[VB]	Public	Sub	VaryByUnspecifiedParameters()
[JScript]	public	function	VaryByUnspecifiedParameters();

### *Description*

Sets the **Vary** HTTP header to the value \* (an asterisk) and causes all other **Vary** header information to be dropped.

The **Vary** header indicates the request-header fields that the server uses to determine which of multiple cached responses is sent in response to a client request. The \* field specifies that the server selects the response based on parameters not specified in request headers (for example, the network address of the client).

HttpCacheVaryByParams class (System.Web)

### VaryByUnspecifiedParameters

### *Description*

Indicates that a cache should contain multiple representations (cached responses) for a particular URI. This class is an encapsulation that provides a type-safe way to set the **Vary** HTTP header.

For more information on HTTP cache control headers, see RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1, available on the World Wide Web Consortium's Web site at <http://www.w3c.org>. See section 14, "Header Field Definitions", for complete details.

IgnoreParams

VaryByUnspecifiedParameters

[C#]            public            bool            IgnoreParams            {get;            set;}

[C++] public: \_\_property bool get\_IgnoreParams();public: \_\_property void  
set\_IgnoreParams(bool);

[VB]            Public            Property            IgnoreParams            As            Boolean

[JScript] public function get IgnoreParams() : Boolean;public function set  
IgnoreParams(Boolean);

### *Description*

Gets or sets a value indicating whether HTTP header cache control parameters are ignored.

Item

VaryByUnspecifiedParameters

[C#]            public            bool            this[string            header]            {get;            set;}

[C++] public: \_\_property bool get\_Item(String\* header);public: \_\_property void  
set\_Item(String\*            header,            bool);

[VB] Public Default Property Item(ByVal header As String) As Boolean

[JScript]            returnValue            =

```

1  HttpCacheVaryByParamsObject.Item(header);HttpCacheVaryByParamsObject.It
2  m(header)                                =                                returnValue;
3

```

#### *Description*

Gets or sets the name of the cache-control header that is used to select one of several different cached responses. The name of the custom header.

HttpClientCertificate class (System.Web)

ToString

#### *Description*

The **HttpClientCertificate** collection retrieves the certification fields (specified in the X.509 standard) from a request issued by the Web browser.

AllKeys

BinaryIssuer

ToString

#### *Description*

CertEncoding

ToString

```

24  [C#]          public          int          CertEncoding          {get;}

```

```

25  [C++]         public:         __property    int          get_CertEncoding();

```

1 [VB] Public ReadOnly Property CertEncoding As Integer  
 2 [JScript] public function get CertEncoding() : int;

3

4 *Description*

5

6 Certificate

7 ToString

8

9 [C#] public byte[] Certificate {get;}

10 [C++] public: \_\_property unsigned char get\_Certificate();

11 [VB] Public ReadOnly Property Certificate As Byte ()

12 [JScript] public function get Certificate() : Byte[];

13

14 *Description*

15 A string containing the binary stream of the entire certificate content in

16 ASN.1 format.

17 Cookie

18 ToString

19

20 [C#] public string Cookie {get;}

21 [C++] public: \_\_property String\* get\_Cookie();

22 [VB] Public ReadOnly Property Cookie As String

23 [JScript] public function get Cookie() : String;

24

25 *Description*



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

Count

Flags

ToString

*Description*

A set of flags that provide additional client certificate information.

IsPresent

ToString

[C#]	public	bool	IsPresent	{get;}
[C++]	public:	__property	bool	get_IsPresent();
[VB]	Public	ReadOnly	Property	IsPresent As Boolean
[JScript]	public	function	get	IsPresent() : Boolean;

*Description*

IsReadOnly

Issuer

ToString

*Description*

1 A string that contains a list of subfield values containing information about  
2 the certificate issuer.

3 IsValid

4 ToString

5  
6 [C#] public bool IsValid {get;}

7 [C++] public: \_\_property bool get\_IsValid();

8 [VB] Public ReadOnly Property IsValid As Boolean

9 [JScript] public function get IsValid() : Boolean;

10  
11 *Description*

12  
13 Item

14 Item

15 Keys

16 KeySize

17 ToString

18  
19  
20 *Description*

21  
22 PublicKey

23 ToString

24  
25 [C#] public byte[] PublicKey {get;}

```

1 [C++] public: __property unsigned char get_PublicKey();
2 [VB] Public ReadOnly Property PublicKey As Byte ()
3 [JScript] public function get PublicKey() : Byte[];
4

```

#### *Description*

SecretKeySize

ToString

```

10 [C#] public int SecretKeySize {get;}
11 [C++] public: __property int get_SecretKeySize();
12 [VB] Public ReadOnly Property SecretKeySize As Integer
13 [JScript] public function get SecretKeySize() : int;
14

```

#### *Description*

SerialNumber

ToString

```

20 [C#] public string SerialNumber {get;}
21 [C++] public: __property String* get_SerialNumber();
22 [VB] Public ReadOnly Property SerialNumber As String
23 [JScript] public function get SerialNumber() : String;
24

```

#### *Description*

1 A string that contains the certification serial number as an ASCII  
2 representation of hexadecimal bytes separated by hyphens (-). For example, 04-67-  
3 F3-02.

4 ServerIssuer

5 ToString

6  
7 [C#] public string ServerIssuer {get;}

8 [C++] public: \_\_property String\* get\_ServerIssuer();

9 [VB] Public ReadOnly Property ServerIssuer As String

10 [JScript] public function get ServerIssuer() : String;

11  
12 *Description*

13  
14 ServerSubject

15 ToString

16  
17 [C#] public string ServerSubject {get;}

18 [C++] public: \_\_property String\* get\_ServerSubject();

19 [VB] Public ReadOnly Property ServerSubject As String

20 [JScript] public function get ServerSubject() : String;

21  
22 *Description*

23  
24 Subject

25 ToString

[C#]	public	string	Subject	{get;}
[C++]	public:	__property	String*	get_Subject();
[VB]	Public	ReadOnly	Property	Subject As String
[JScript]	public	function	get Subject()	: String;

### Description

A string that contains a list of subfield values that contain information about the subject of the certificate. If this value is specified without a *SubField*, the ClientCertificate collection returns a comma-separated list of subfields. For example, C=US, O=Msft, and so on.

ValidFrom

ToString

[C#]	public	DateTime	ValidFrom	{get;}
[C++]	public:	__property	DateTime	get_ValidFrom();
[VB]	Public	ReadOnly	Property	ValidFrom As DateTime
[JScript]	public	function	get ValidFrom()	: DateTime;

### Description

A date specifying when the certificate becomes valid. This date varies with international settings.

ValidUntil

ToString

```

1
2 [C#]      public      DateTime      ValidUntil      {get;}
3 [C++]     public:     __property     DateTime      get_ValidUntil();
4 [VB]      Public     ReadOnly     Property     ValidUntil     As     DateTime
5 [JScript] public     function     get     ValidUntil()     :     DateTime;
6

```

### 7 *Description*

8 A date specifying when the certificate expires. The year value is displayed  
9 as a four-digit number.

### 10 *Get*

```

11
12 [C#]      public      override     string      Get(string      field);
13 [C++]     public:     String*      Get(String*      field);
14 [VB]      Overrides Public Function Get(ByVal field As String) As String
15 [JScript] public     override     function     Get(field : String) : String;
16

```

### 17 *Description*

18 Allows access to individual items in the collection by name. The name of  
19 the item in the collection to retrieve.

20 HttpCompileException class (System.Web)

### 21 *ToString*

### 24 *Description*

25 The exception that is thrown when a compiler error occurs.

HttpCompileException

*Example Syntax:*

ToString

[C#] public HttpCompileException(CompilerResults results, string sourceCode);

[C++] public: HttpCompileException(CompilerResults\* results, String\* sourceCode);

[VB] Public Sub New(ByVal results As CompilerResults, ByVal sourceCode As String)

[JScript] public function HttpCompileException(results : CompilerResults, sourceCode : String);

### *Description*

Initializes a new instance of the **System.Web.HttpCompileException** class. A **System.CodeDom.Compiler.CompilerResults** containing compiler output and error information. The name of the file being compiled when the error occurs.

ErrorCode

HelpLink

HResult

InnerException

Message

Results

ToString

1  
2  
3 *Description*

4 Gets compiler output and error information for the exception.

5 Source

6 SourceCode

7 ToString

8  
9  
10 *Description*

11 Gets the name of the source file being compiled when the error occurs.

12 StackTrace

13 TargetSite

14 HttpContext class (System.Web)

15 ToString

16  
17  
18 *Description*

19 Encapsulates all HTTP-specific information about an individual HTTP  
20 request.

21 Classes that inherit the **System.Web.IHttpModule** and  
22 **System.Web.IHttpHandler** interfaces are provided a reference to an  
23 **HttpContext** object for the current HTTP request. The object provides access to  
24 the intrinsic **System.Web.HttpContext.Request**,  
25



**System.Web.HttpContext.Response** , and **System.Web.HttpContext.Server** objects for the request.

**HttpContext**

*Example Syntax:*

**ToString**

[C#]            public            HttpContext(HttpContextWorkerRequest            wr);

[C++]           public:            HttpContext(HttpContextWorkerRequest\*            wr);

[VB]    Public    Sub    New(ByVal    wr    As    HttpContextWorkerRequest)

[JScript]    public    function    HttpContext(wr    :    HttpContextWorkerRequest);

### *Description*

Initializes a new instance of the **System.Web.HttpContext** class. The **System.Web.HttpContextWorkerRequest** object for the current HTTP request.

**HttpContext**

*Example Syntax:*

**ToString**

[C#]    public    HttpContext(HttpContextRequest    request,    HttpContextResponse    response);

[C++]    public:    HttpContext(HttpContextRequest\*    request,    HttpContextResponse\*    response);

[VB]    Public    Sub    New(ByVal    request    As    HttpContextRequest,    ByVal    response    As    HttpContextResponse)

[JScript]    public    function    HttpContext(request    :    HttpContextRequest,    response    :    HttpContextResponse); Initializes a new instance of the **System.Web.HttpContext** class.

## Description

Initializes a new instance of the **System.Web.HttpContext** class. The **System.Web.HttpRequest** object for the current HTTP request. The **System.Web.HttpResponse** object for the current HTTP request.

AllErrors

ToString

```
[C#]      public      Exception[]      AllErrors      {get;}
```

```
[C++]     public:     __property      Exception*      get_AllErrors();
```

```
[VB]      Public      ReadOnly      Property      AllErrors      As      Exception      ()
```

```
[JScript] public      function      get      AllErrors()      :      Exception[];
```

## Description

Gets an array of errors accumulated while processing an HTTP request.

Application

ToString

```
[C#]      public      HttpApplicationState      Application      {get;}
```

```
[C++]     public:     __property      HttpApplicationState*      get_Application();
```

```
[VB]      Public      ReadOnly      Property      Application      As      HttpApplicationState
```

```
[JScript] public      function      get      Application()      :      HttpApplicationState;
```

## Description

1 Gets the **System.Web.HttpApplicationState** object for the current HTTP  
2 request.

3 ApplicationInstance

4 ToString

5  
6 [C#] public HttpApplication ApplicationInstance {get; set;}

7 [C++] public: \_\_property HttpApplication\* get\_ApplicationInstance();public:

8 \_\_property void set\_ApplicationInstance(HttpApplication\*);

9 [VB] Public Property ApplicationInstance As HttpApplication

10 [JScript] public function get ApplicationInstance() : HttpApplication;public

11 function set ApplicationInstance(HttpApplication);

12  
13 *Description*

14 Gets or sets the **System.Web.HttpApplicationState** object for the current  
15 HTTP request.

16 Cache

17 ToString

18  
19 [C#] public Cache Cache {get;}

20 [C++] public: \_\_property Cache\* get\_Cache();

21 [VB] Public ReadOnly Property Cache As Cache

22 [JScript] public function get Cache() : Cache;

23  
24 *Description*

25 Gets the **System.Web.Caching.Cache** object for the current HTTP request.

1 Current  
2 ToString  
3  
4 [C#] public static HttpContext Current {get;}  
5 [C++] public: \_\_property static HttpContext\* get\_Current();  
6 [VB] Public Shared ReadOnly Property Current As HttpContext  
7 [JScript] public static function get Current() : HttpContext;

8  
9 *Description*

10 Gets the **System.Web.HttpContext** object for the current HTTP request.

11 Error

12 ToString  
13

14 [C#] public Exception Error {get;}  
15 [C++] public: \_\_property Exception\* get\_Error();  
16 [VB] Public ReadOnly Property Error As Exception  
17 [JScript] public function get Error() : Exception;

18  
19 *Description*

20 Gets the first error (if any) accumulated during HTTP request processing.

21 Handler

22 ToString  
23

24 [C#] public IHttpHandler Handler {get; set;}  
25 [C++] public: \_\_property IHttpHandler\* get\_Handler();public: \_\_property void

```

1 set_Handler(IHttpHandler*);
2 [VB]      Public      Property      Handler      As      IHttpHandler
3 [JScript] public function get Handler() : IHttpHandler;public function set
4 Handler(IHttpHandler);
5

```

#### *Description*

Gets or sets the **System.Web.IHttpHandler** object for the current HTTP request.

IsCustomErrorEnabled

ToString

```

12 [C#]      public      bool      IsCustomErrorEnabled      {get;}
13 [C++]      public:      __property      bool      get_IsCustomErrorEnabled();
14 [VB]      Public      ReadOnly      Property      IsCustomErrorEnabled      As      Boolean
15 [JScript]      public      function      get      IsCustomErrorEnabled()      :      Boolean;
16

```

#### *Description*

Gets a value indicating whether custom errors are enabled for the current HTTP request.

IsDebuggingEnabled

ToString

```

23 [C#]      public      bool      IsDebuggingEnabled      {get;}
24 [C++]      public:      __property      bool      get_IsDebuggingEnabled();
25 [VB]      Public      ReadOnly      Property      IsDebuggingEnabled      As      Boolean

```

1 [JScript] public function get IsDebugEnabled() : Boolean;

2

3 *Description*

4 Gets a value indicating whether the current HTTP request is in debug  
5 mode.

6 Items

7 ToString

8

9 [C#] public IDictionary Items {get;}

10 [C++] public: \_\_property IDictionary\* get\_Items();

11 [VB] Public ReadOnly Property Items As IDictionary

12 [JScript] public function get Items() : IDictionary;

13

14 *Description*

15 Gets a key-value collection that can be used to organize and share data  
16 between an **System.Web.IHttpModule** and an **System.Web.IHttpHandler**  
17 during an HTTP request.

18 Request

19 ToString

20

21 [C#] public HttpRequest Request {get;}

22 [C++] public: \_\_property HttpRequest\* get\_Request();

23 [VB] Public ReadOnly Property Request As HttpRequest

24 [JScript] public function get Request() : HttpRequest;

25

1  
2 *Description*

3 Gets the **System.Web.HttpRequest** object for the current HTTP request.

4 Response

5 ToString

6  
7 [C#] public HttpResponseMessage Response {get;}

8 [C++] public: \_\_property HttpResponseMessage\* get\_Response();

9 [VB] Public ReadOnly Property Response As HttpResponseMessage

10 [JScript] public function get Response() : HttpResponseMessage;

11  
12 *Description*

13 Gets the **System.Web.HttpResponse** object for the current HTTP  
14 response.

15 Server

16 ToString

17  
18 [C#] public HttpServerUtility Server {get;}

19 [C++] public: \_\_property HttpServerUtility\* get\_Server();

20 [VB] Public ReadOnly Property Server As HttpServerUtility

21 [JScript] public function get Server() : HttpServerUtility;

22  
23 *Description*

24 Gets the **System.Web.HttpServerUtility** object that provides methods  
25 used in processing Web requests.

Session

ToString

[C#] public HttpSessionState Session {get;}

[C++] public: \_\_property HttpSessionState\* get\_Session();

[VB] Public ReadOnly Property Session As HttpSessionState

[JScript] public function get Session() : HttpSessionState;

#### *Description*

Gets the **System.Web.SessionState** instance for the current HTTP request.

SkipAuthorization

ToString

[C#] public bool SkipAuthorization {get; set;}

[C++] public: \_\_property bool get\_SkipAuthorization();public: \_\_property void set\_SkipAuthorization(bool);

[VB] Public Property SkipAuthorization As Boolean

[JScript] public function get SkipAuthorization() : Boolean;public function set SkipAuthorization(Boolean);

#### *Description*

Gets or sets a value that specifies whether the URLAuthorization module will skip the authorization check for the current request.

**SkipAuthorization** is for advanced use by authentication modules that need to redirect to an anonymous-allowed page. The Forms authentication module



and Passport authentication module both set this property when redirecting to a configured login page. Setting this requires the **ControlPrincipal** flag to be set in **System.Security.Permissions.SecurityPermission.Flags** .

Timestamp

ToString

[C#]            public            DateTime            Timestamp            {get;}

[C++]           public:            \_\_property            DateTime            get\_Timestamp();

[VB]    Public    ReadOnly    Property    Timestamp    As    DateTime

[JScript]    public    function    get    Timestamp()    :    DateTime;

#### *Description*

Gets the initial timestamp of the current HTTP request.

Trace

ToString

[C#]            public            TraceContext            Trace            {get;}

[C++]           public:            \_\_property            TraceContext\*            get\_Trace();

[VB]    Public    ReadOnly    Property    Trace    As    TraceContext

[JScript]    public    function    get    Trace()    :    TraceContext;

#### *Description*

Gets the **System.Web.TraceContext** object for the current HTTP response.

User

ToString

```

1
2 [C#]      public      IPrincipal      User      {get;      set;}
3 [C++]  public:  __property  IPrincipal*  get_User();public:  __property  void
4 set_User(IPrincipal*);
5 [VB]      Public      Property      User      As      IPrincipal
6 [JScript] public  function  get  User()  :  IPrincipal;public  function  set
7 User(IPrincipal);
8

```

### *Description*

Gets or sets security information for the current HTTP request.

Setting this property requires the **ControlPrincipal** flag to be set in **System.Security.Permissions.SecurityPermission.Flags**.

### **AddError**

```

15 [C#]      public      void      AddError(Exception      errorInfo);
16 [C++]      public:      void      AddError(Exception*      errorInfo);
17 [VB]      Public      Sub      AddError(ByVal      errorInfo      As      Exception)
18 [JScript] public  function  AddError(errorInfo      :      Exception);
19

```

### *Description*

Adds an exception to the exception collection for the current HTTP request.

The **System.Exception** object to add to the exception collection.

### **ClearError**

```

25 [C#]      public      void      ClearError();

```

1	[C++]	public:	void	ClearError();
2	[VB]	Public	Sub	ClearError()
3	[JScript]	public	function	ClearError();

4

5 *Description*

6 Clears all errors for the current HTTP request.

7 **GetAppConfig**

8

9	[C#]	public	static	object	GetAppConfig(string	name);
10	[C++]	public:	static	Object*	GetAppConfig(String*	name);
11	[VB]	Public	Shared	Function	GetAppConfig(ByVal name As String)	As Object
12	[JScript]	public	static	function	GetAppConfig(name : String)	: Object;

13

14 *Description*

15 Returns requested configuration information for the current application The

16 application configuration tag that information is requested for.

17 **GetConfig**

18

19	[C#]	public	object	GetConfig(string	name);
20	[C++]	public:	Object*	GetConfig(String*	name);
21	[VB]	Public	Function	GetConfig(ByVal name As String)	As Object
22	[JScript]	public	function	GetConfig(name : String)	: Object;

23 configuration information for the current HTTP request.

24

25 *Description*

1 Returns requested configuration information for the current HTTP request.  
2 The configuration tag that information is requested for.

### 3 RewritePath

4  
5 [C#] public void RewritePath(string path);  
6 [C++] public: void RewritePath(String\* path);  
7 [VB] Public Sub RewritePath(ByVal path As String)  
8 [JScript] public function RewritePath(path : String);  
9

### 10 *Description*

11 Assigns an internal rewrite path. The internal rewrite path.

12 IServiceProvider.GetService

13  
14 [C#] object IServiceProvider.GetService(Type service);  
15 [C++] Object\* IServiceProvider::GetService(Type\* service);  
16 [VB] Function GetService(ByVal service As Type) As Object Implements  
17 IServiceProvider.GetService  
18 [JScript] function IServiceProvider.GetService(service : Type) : Object;

19 HttpCookie class (System.Web)

20 ToString

### 23 *Description*

24 Provides a type-safe way to create and manipulate individual HTTP  
25 cookies.

The **System.Web.HttpCookie** class gets and sets properties of individual cookies. The **System.Web.HttpCookieCollection** class provides methods to store, retrieve, and manage all the cookies for an entire Web application. ASP.NET code uses the intrinsic **System.Web.HttpResponse.Cookies** object to create cookies and add them to the cookie collection. When delivering a Web page to a client, the server sends the entire cookie collection with the **Set-Cookie** header.

**HttpCookie**

*Example Syntax:*

**ToString**

[C#]	public	HttpCookie(string	name);
[C++]	public:	HttpCookie(String*	name);
[VB]	Public Sub	New(ByVal name	As String)
[JScript]	public function	HttpCookie(name : String);	Initializes a new instance of
the		<b>System.Web.HttpCookie</b>	class.

*Description*

Creates and names a new cookie. The name of the new cookie.

**HttpCookie**

*Example Syntax:*

**ToString**

[C#]	public	HttpCookie(string	name,	string	value);
[C++]	public:	HttpCookie(String*	name,	String*	value);

```

1 [VB] Public Sub New(ByVal name As String, ByVal value As String)
2 [JScript] public function HttpCookie(name : String, value : String);

```

#### *Description*

Creates, names, and assigns a value to a new cookie. The name of the new cookie. The value of the new cookie.

Domain

ToString

```

10 [C#]          public          string          Domain          {get;          set;}

```

```

11 [C++] public: __property String* get_Domain();public: __property void
12 set_Domain(String*);

```

```

13 [VB]          Public          Property          Domain          As          String

```

```

14 [JScript] public function get Domain() : String;public function set Domain(String);

```

#### *Description*

Gets or sets the domain to associate the cookie with.

Setting the **Domain** attribute limits transmission of the cookie to clients requesting a resource from that domain.

Expires

ToString

```

23 [C#]          public          DateTime          Expires          {get;          set;}

```

```

24 [C++] public: __property DateTime get_Expires();public: __property void
25 set_Expires(DateTime);

```

```

1  [VB]          Public          Property      Expires      As          DateTime
2  [JScript] public function get Expires() : DateTime;public function set
3  Expires(DateTime);

```

#### *Description*

Gets or sets the expiration date and time for the cookie.

HasKeys

ToString

```

10 [C#]          public          bool          HasKeys          {get;}
11 [C++] public:      __property      bool          get_HasKeys();
12 [VB] Public      ReadOnly      Property      HasKeys      As      Boolean
13 [JScript] public      function      get      HasKeys()      :      Boolean;

```

#### *Description*

Gets a value indicating whether a cookie has subkeys.

Item

ToString

```

20 [C#]          public          string      this[string      key]      {get;      set;}
21 [C++] public: __property String* get_Item(String* key);public: __property void
22 set_Item(String*          key,          String*);
23 [VB] Public Default Property Item(ByVal key As String) As String
24 [JScript] returnValue = HttpCookieObject.Item(key);HttpCookieObject.Item(key)
25 =          returnValue;

```

1  
2 *Description*

3       Shortcut for **HttpCookie.Values[ key ]**. This property is provided for  
4 compatibility with previous versions of ASP. Key (index) of cookie value.

5       Name

6       ToString

7  
8 [C#]       public       string       Name       {get;       set;}

9 [C++]   public:   \_\_property   String\*   get\_Name();public:   \_\_property   void  
10 set\_Name(String\*);

11 [VB]       Public       Property       Name       As       String

12 [JScript] public function get Name() : String;public function set Name(String);

13  
14 *Description*

15       Gets or sets the name of a cookie.

16       Path

17       ToString

18  
19 [C#]       public       string       Path       {get;       set;}

20 [C++]   public:   \_\_property   String\*   get\_Path();public:   \_\_property   void  
21 set\_Path(String\*);

22 [VB]       Public       Property       Path       As       String

23 [JScript] public function get Path() : String;public function set Path(String);

24  
25 *Description*



Gets or sets the virtual path to transmit with the current cookie.

The **Path** property extends the **Domain** property to completely describe the specific URL that the cookie applies to. For example, in the URL `http://www.microsoft.com/asp`, the domain is `www.microsoft.com` and the path is `/asp`.

Secure

ToString

```
[C#]          public          bool          Secure          {get;          set;}
```

```
[C++] public:  __property  bool  get_Secure();public:  __property  void  
set_Secure(bool);
```

```
[VB]          Public          Property          Secure          As          Boolean
```

```
[JScript] public function get Secure() : Boolean;public function set  
Secure(Boolean);
```

#### *Description*

Gets or sets a value indicating whether to transmit the cookie securely (that is, over HTTPS only).

Value

ToString

```
[C#]          public          string          Value          {get;          set;}
```

```
[C++] public:  __property  String*  get_Value();public:  __property  void  
set_Value(String*);
```

```
[VB]          Public          Property          Value          As          String
```

1 [JScript] public function get Value() : String;public function set Value(String);

2  
3 *Description*

4 Gets or sets an individual cookie value.

5 Values

6 ToString

7  
8 [C#] public NameValueCollection Values {get;}

9 [C++] public: \_\_property NameValueCollection\* get\_Values();

10 [VB] Public ReadOnly Property Values As NameValueCollection

11 [JScript] public function get Values() : NameValueCollection;

12  
13 *Description*

14 Gets a collection of key-and-value value pairs that are contained within a  
15 single cookie object.

16 HttpCookieCollection class (System.Web)

17 ToString

18  
19  
20 *Description*

21 Provides a type-safe way to manipulate HTTP cookies.

22 HttpCookieCollection

23 *Example Syntax:*

24 ToString

1				
2	[C#]	public		HttpCookieCollection();
3	[C++]	public:		HttpCookieCollection();
4	[VB]	Public	Sub	New()
5	[JScript]	public	function	HttpCookieCollection();

6

7 *Description*

8       Initializes a new instance of the **System.Web.HttpCookieCollection** class.

9       ASP.NET includes two intrinsic cookie collections. The collection  
 10 accessible through **System.Web.HttpRequest.Cookies** contains cookies  
 11 transmitted by the client to the server in the **Cookie** header. The collection  
 12 accessible through **System.Web.HttpResponse.Cookies** contains cookies  
 13 generated on the server and transmitted to the client in the **Set-Cookie** header.

14       AllKeys

15       ToString

16							
17	[C#]	public	string[]	AllKeys		{get;}	
18	[C++]	public:	__property	String*		get_AllKeys();	
19	[VB]	Public	ReadOnly Property	AllKeys	As	String	()
20	[JScript]	public	function	get	AllKeys()	:	String[];

21

22 *Description*

23       Gets a string array containing all the keys (cookie names) in the cookie  
 24 collection.

25       Count

1       IsReadOnly

2       Item

3       ToString

4

5

6       *Description*

7           Gets the cookie with the specified numerical index from the cookie  
8 collection. The index of the cookie to retrieve from the collection.

9       Item

10      ToString

11

12   [C#]       public       HttpCookie       this[string       name]       {get;}

13   [C++]   public:   \_\_property   HttpCookie\*   get\_Item(String\*   name);

14   [VB]   Public   Default   ReadOnly   Property   Item(ByVal   name   As   String)   As  
15   HttpCookie

16   [JScript]   returnValue = HttpCookieCollectionObject.Item(name); Gets the cookie  
17 with the specified name from the cookie collection. This property is overloaded to  
18 allow retrieval of cookies by either name or numerical index.

19

20       *Description*

21           Gets the cookie with the specified name from the cookie collection. Name  
22 of cookie to retrieve.

23       Keys

24       Add

25

```

1
2 [C#]      public      void      Add(HttpCookie      cookie);
3 [C++]     public:     void      Add(HttpCookie*      cookie);
4 [VB]      Public      Sub      Add(ByVal      cookie      As      HttpCookie)
5 [JScript] public      function Add(cookie      :      HttpCookie);

```

### 6 7 *Description*

8 Adds the specified cookie to the cookie collection.

9 Any number of cookie collections can exist within an application, but only  
10 the collection referenced by the intrinsic **System.Web.HttpResponse.Cookies**  
11 object is sent to the client. The **System.Web.HttpCookie** to add to the collection.

### 12 *Clear*

```

13
14 [C#]      public      void      Clear();
15 [C++]     public:     void      Clear();
16 [VB]      Public      Sub      Clear()
17 [JScript] public      function Clear();

```

### 18 19 *Description*

20 Clears all cookies from the cookie collection.

### 21 *CopyTo*

```

22
23 [C#]      public      void      CopyTo(Array      dest,      int      index);
24 [C++]     public:     void      CopyTo(Array*      dest,      int      index);
25 [VB]      Public      Sub      CopyTo(ByVal dest As Array, ByVal index As Integer)

```

1 [JScript] public function CopyTo(dest : Array, index : int);

2  
3 *Description*

4 Copies members of the cookie collection to an **System.Array** beginning at  
5 the specified index of the array. The destination **System.Array**. The index of the  
6 destination array where copying starts.

7 Get

8  
9 [C#] public HttpCookie Get(int index);

10 [C++] public: HttpCookie\* Get(int index);

11 [VB] Public Function Get(ByVal index As Integer) As HttpCookie

12 [JScript] public function Get(index : int) : HttpCookie;

13  
14 *Description*

15 Returns the **System.Web.HttpCookie** item with the specified index from  
16 the cookie collection. The index of the cookie to return from the collection.

17 Get

18  
19 [C#] public HttpCookie Get(string name);

20 [C++] public: HttpCookie\* Get(String\* name);

21 [VB] Public Function Get(ByVal name As String) As HttpCookie

22 [JScript] public function Get(name : String) : HttpCookie; Returns an individual

23 **System.Web.HttpCookie** object from the cookie collection. This property is  
24 overloaded to allow retrieval of cookies by either name or numerical index.

## Description

Returns the **System.Web.HttpCookie** item with the specified name from the cookie collection.

If the named cookie does not exist, this method creates a new cookie with that name. The name of the cookie to retrieve from the collection.

### GetKey

[C#]	public	string	GetKey(int	index);
[C++]	public:	String*	GetKey(int	index);
[VB]	Public	Function	GetKey(ByVal index As Integer)	As String
[JScript]	public	function	GetKey(index : int)	: String;

## Description

Returns the key (name) of the cookie at the specified numerical index. The index of the key to retrieve from the collection.

### Remove

[C#]	public	void	Remove(string	name);
[C++]	public:	void	Remove(String*	name);
[VB]	Public	Sub	Remove(ByVal name As String)	
[JScript]	public	function	Remove(name : String);	

## Description

1 Removes the cookie with the specified name from the collection. The name  
2 of the cookie to remove from the collection.

3 Set

4  
5 [C#] public void Set(HttpCookie cookie);

6 [C++] public: void Set(HttpCookie\* cookie);

7 [VB] Public Sub Set(ByVal cookie As HttpCookie)

8 [JScript] public function Set(cookie : HttpCookie);

9  
10 *Description*

11 Updates the value of an existing cookie in a cookie collection. The  
12 **System.Web.HttpCookie** object to update.

13 HttpException class (System.Web)

14 ToString

15  
16  
17 *Description*

18 Provides a means of generating HTTP exceptions.

19 HttpException

20 *Example Syntax:*

21 ToString

22  
23 [C#] public HttpException();

24 [C++] public: HttpException();

25 [VB] Public Sub New()



1 [JScript] public function HttpException(); Constructs a new **System.Exception**  
2 object.

3  
4 *Description*

5 Constructs an empty **Exception** object.

6 When handling exceptions, it is sometimes useful to capture a series of  
7 related exceptions with the outer exceptions being thrown in response to an inner  
8 exceptions.

9 HttpException

10 *Example Syntax:*

11 ToString

12  
13 [C#] public HttpException(string message);

14 [C++] public: HttpException(String\* message);

15 [VB] Public Sub New(ByVal message As String)

16 [JScript] public function HttpException(message : String);

17  
18 *Description*

19 Constructs an **System.Exception** using a supplied error message. The  
20 message displayed to the client when the exception is thrown.

21 HttpException

22 *Example Syntax:*

23 ToString

24  
25 [C#] public HttpException(int httpCode, string message);

```

1 [C++] public: HttpException(int httpCode, String* message);
2 [VB] Public Sub New(ByVal httpCode As Integer, ByVal message As String)
3 [JScript] public function HttpException(httpCode : int, message : String);
4

```

#### *Description*

Constructs an **System.Exception** using an HTTP error code and an error message. The HTTP error code displayed on the client. The message displayed to the client when the exception is thrown.

HttpException

*Example Syntax:*

ToString

```

13 [C#] public HttpException(string message, Exception innerException);
14 [C++] public: HttpException(String* message, Exception* innerException);
15 [VB] Public Sub New(ByVal message As String, ByVal innerException As
16 Exception)
17 [JScript] public function HttpException(message : String, innerException :
18 Exception);
19

```

#### *Description*

Constructs an **System.Exception** using an error message and the **System.Exception.InnerException** property.

When handling exceptions, it is sometimes useful to capture a series of related exceptions with the outer exceptions being thrown in response to an inner

exception. The message displayed to the client when the exception is thrown. The **System.Exception.InnerException**, if any, that threw the current exception.

HttpException

*Example Syntax:*

ToString

[C#] public HttpException(string message, int hr);

[C++] public: HttpException(String\* message, int hr);

[VB] Public Sub New(ByVal message As String, ByVal hr As Integer)

[JScript] public function HttpException(message : String, hr : int);

### Description

Constructs an **System.Exception** using error message and an exception code. The error message displayed to the client when the exception is thrown. The exception code that defines the error.

HttpException

*Example Syntax:*

ToString

[C#] public HttpException(int httpCode, string message, Exception innerException);

[C++] public: HttpException(int httpCode, String\* message, Exception\* innerException);

[VB] Public Sub New(ByVal httpCode As Integer, ByVal message As String, ByVal innerException As Exception)

1 [JScript] public function HttpException(httpCode : int, message : String,  
2 innerException : Exception);

3  
4 *Description*

5 Constructs an **System.Exception** using an HTTP error code, an error  
6 message, and the **System.Exception.InnerException** property.

7 When handling exceptions, it is sometimes useful to capture a series of  
8 related exceptions with the outer exceptions being thrown in response to an inner  
9 exceptions. The HTTP error code displayed to the client. The message displayed  
10 to the client. The **InnerException** , if any, that threw the current exception.

11 HttpException

12 *Example Syntax:*

13 ToString

14  
15 [C#] public HttpException(int httpCode, string message, int hr);

16 [C++] public: HttpException(int httpCode, String\* message, int hr);

17 [VB] Public Sub New(ByVal httpCode As Integer, ByVal message As String,

18 ByVal hr As Integer)

19 [JScript] public function HttpException(httpCode : int, message : String, hr : int);

20  
21 *Description*

22 Constructs an **System.Exception** using HTTP error code, an error message,  
23 and an exception code. The HTTP error code displayed on the client. The error  
24 message displayed to the client. The error code that defines the error.

25 ErrorCode

1       HelpLink  
 2       HResult  
 3       InnerException  
 4       Message  
 5       Source  
 6       StackTrace  
 7       TargetSite  
 8       CreateFromLastError

9  
 10   [C#]   public   static   HttpException   CreateFromLastError(string   message);  
 11   [C++]   public:   static   HttpException\*   CreateFromLastError(String\*   message);  
 12   [VB]   Public Shared Function CreateFromLastError(ByVal message As String) As  
 13   HttpException  
 14   [JScript]   public   static   function   CreateFromLastError(message : String) :  
 15   HttpException;

16  
 17   *Description*

18       Creates a new **System.Exception** based on the previous **Exception** .  
 19   *Return Value:* An **Exception** with the same error identification code as the  
 20   previous **Exception** but with a new message. The message to be displayed to the  
 21   client when the exception is thrown.

22       GetHtmlErrorMessage

23  
 24   [C#]               public               string               GetHtmlErrorMessage();  
 25   [C++]              public:              String\*             GetHtmlErrorMessage();



Clients encode files and transmit them in the content body using multipart MIME format with an HTTP **Content-Type** header of **multipart/form-data** . ASP.NET extracts the encoded file(s) from the content body into individual members of an **System.Web.HttpFileCollection** . Methods and properties of the **System.Web.HttpPostedFile** class provide access to the contents and properties of each file.

AllKeys

ToString

[C#]            public            string[]            AllKeys            {get;}

[C++]            public:            \_\_property            String\*            get\_AllKeys();

[VB]    Public    ReadOnly    Property    AllKeys    As    String    ()

[JScript]    public    function    get    AllKeys()    :    String[];

#### *Description*

Gets a string array containing the keys (names) of all members in the file collection.

Count

IsReadOnly

Item

ToString

#### *Description*

1 Gets the object with the specified numerical index from the  
2 **System.Web.HttpFileCollection** . The index of the item to get from the file  
3 collection.

4 Item

5 ToString

6  
7 [C#] public HttpPostedFile this[string name] {get;}

8 [C++] public: \_\_property HttpPostedFile\* get\_Item(String\* name);

9 [VB] Public Default ReadOnly Property Item(ByVal name As String) As  
10 HttpPostedFile

11 [JScript] returnValue = HttpFileCollectionObject.Item(name); Gets an individual  
12 **System.Web.HttpPostedFile** object from the file collection. This property is  
13 overloaded to allow retrieval of objects by either name or numerical index.

14  
15 *Description*

16 Gets the object with the specified name from the file collection. Name of  
17 item to be returned.

18 Keys

19 CopyTo

20  
21 [C#] public void CopyTo(Array dest, int index);

22 [C++] public: void CopyTo(Array\* dest, int index);

23 [VB] Public Sub CopyTo(ByVal dest As Array, ByVal index As Integer)

24 [JScript] public function CopyTo(dest : Array, index : int);

25



## Description

Copies members of the file collection to an **System.Array** beginning at the specified index of the array. The destination **System.Array**. The index of the destination array where copying starts.

### Get

```
[C#]      public      HttpPostedFile      Get(int      index);
[C++]      public:      HttpPostedFile*      Get(int      index);
[VB] Public Function Get(ByVal index As Integer) As HttpPostedFile
[JScript] public function Get(index : int) : HttpPostedFile;
```

## Description

Returns the **System.Web.HttpPostedFile** object with the specified numerical index from the file collection. The index of the object to be returned from the file collection.

### Get

```
[C#]      public      HttpPostedFile      Get(string      name);
[C++]      public:      HttpPostedFile*      Get(String*      name);
[VB] Public Function Get(ByVal name As String) As HttpPostedFile
[JScript] public function Get(name : String) : HttpPostedFile; Returns an
individual System.Web.HttpPostedFile object from a file collection. This
property is overloaded to allow retrieval of objects by either name or numerical
index.
```

1  
2 *Description*

3 Returns the **System.Web.HttpPostedFile** object with the specified name  
4 from the file collection. The name of the object to be returned from a file  
5 collection.

6 **GetKey**

7  
8 [C#] public string GetKey(int index);  
9 [C++] public: String\* GetKey(int index);  
10 [VB] Public Function GetKey(ByVal index As Integer) As String  
11 [JScript] public function GetKey(index : int) : String;  
12

13 *Description*

14 Returns the name of the **System.Web.HttpFileCollection** member with the  
15 specified numerical index. The index of the object name to be returned.

16 **HttpModuleCollection** class (System.Web)

17 **ToString**  
18  
19

20 *Description*

21 Provides a means of indexing and retrieving a collection of  
22 **System.Web.IHttpModule** objects.

23 **AllKeys**

24 **ToString**  
25

```

1
2 [C#]          public          string[]          AllKeys          {get;}
3 [C++]         public:         __property        String*          get_AllKeys();
4 [VB]   Public  ReadOnly  Property  AllKeys  As  String  ()
5 [JScript]    public    function    get    AllKeys()    :    String[];
6

```

### *Description*

Gets a string array containing all the keys (module names) in the **System.Web.HttpModuleCollection** .

Count

IsReadOnly

Item

ToString

### *Description*

Gets the **System.Web.IHttpModule** object with the specified numerical index from the **System.Web.HttpModuleCollection** . The index of the **System.Web.IHttpModule** object to retrieve from the collection.

Item

ToString

```

23 [C#]          public          IHttpModule        this[string        name]          {get;}
24 [C++]         public:         __property        IHttpModule*        get_Item(String*        name);
25 [VB]   Public  Default  ReadOnly  Property  Item(ByVal  name  As  String)  As

```

1 IHttpModule

2 [JScript] returnValue = HttpModuleCollectionObject.Item(name); Gets the  
3 **System.Web.IHttpModule** object with the specified name from the  
4 **System.Web.HttpModuleCollection** . This property is overloaded to allow  
5 retrieval of modules by either name or numerical index.

6  
7 *Description*

8 Gets the **System.Web.IHttpModule** object with the specified name from  
9 the **System.Web.HttpModuleCollection** . Key of the item to be retrieved.

10 Keys

11 CopyTo

12  
13 [C#] public void CopyTo(Array dest, int index);

14 [C++] public: void CopyTo(Array\* dest, int index);

15 [VB] Public Sub CopyTo(ByVal dest As Array, ByVal index As Integer)

16 [JScript] public function CopyTo(dest : Array, index : int);

17  
18 *Description*

19 Copies members of the module collection to an **System.Array** beginning at  
20 the specified index of the array. The destination **Array**. The index of the  
21 destination **Array** where copying starts.

22 Get

23  
24 [C#] public IHttpModule Get(int index);

25 [C++] public: IHttpModule\* Get(int index);

```

1 [VB] Public Function Get(ByVal index As Integer) As IHttpModule
2 [JScript] public function Get(index : int) : IHttpModule;

```

#### *Description*

Returns the **System.Web.IHttpModule** object with the specified index from the **System.Web.HttpModuleCollection** . Index of the **System.Web.IHttpModule** object to return from the collection.

#### *Get*

```

10 [C#] public IHttpModule Get(string name);

```

```

11 [C++] public: IHttpModule* Get(String* name);

```

```

12 [VB] Public Function Get(ByVal name As String) As IHttpModule

```

```

13 [JScript] public function Get(name : String) : IHttpModule; Returns an individual

```

**System.Web.IHttpModule** object from the **System.Web.HttpModuleCollection**

. This property is overloaded to allow retrieval of modules by either name or

numerical index.

#### *Description*

Returns the **System.Web.IHttpModule** object with the specified name from the **System.Web.HttpModuleCollection** . Key of the item to be retrieved.

#### *GetKey*

```

23 [C#] public string GetKey(int index);

```

```

24 [C++] public: String* GetKey(int index);

```

```

25 [VB] Public Function GetKey(ByVal index As Integer) As String

```

1 [JScript] public function GetKey(index : int) : String;

2  
3 *Description*

4 Returns the key (name) of the **System.Web.IHttpModule** object at the  
5 specified numerical index.. Index of the key to retrieve from the collection.

6 HttpParseException class (System.Web)

7 ToString

8  
9  
10 *Description*

11 The exception that is thrown when a parse error occurs.

12 HttpParseException

13 *Example Syntax:*

14 ToString

15  
16 [C#] public HttpParseException(string message, Exception innerException, string  
17 fileName, int line);

18 [C++] public: HttpParseException(String\* message, Exception\* innerException,  
19 String\* fileName, int line);

20 [VB] Public Sub New(ByVal message As String, ByVal innerException As  
21 Exception, ByVal fileName As String, ByVal line As Integer)

22 [JScript] public function HttpParseException(message : String, innerException :  
23 Exception, fileName : String, line : int);

24  
25 *Description*

1        Initializes a new instance of the **System.Web.HttpParseException** class.  
2        The message displayed to the client when the exception is thrown. The  
3        **System.Exception**, if any, that threw the current exception. The name of the file  
4        being parsed when the error occurs. The number of the line being parsed when the  
5        error occurs.

6        ErrorCode

7        FileName

8        ToString

9  
10  
11        *Description*

12        Gets the name of the file being parsed when the error occurs.

13        HelpLink

14        HResult

15        InnerException

16        Line

17        ToString

18  
19  
20        *Description*

21        Gets the number of the line being parsed when the error occurs.

22        Message

23        Source

24        StackTrace

25        TargetSite

1       HttpPostedFile class (System.Web)

2       ToString

3  
4  
5       *Description*

6       Provides a way to access individual files that have been uploaded by a  
7       client.

8       The **System.Web.HttpFileCollection** class provides access to all the files  
9       uploaded from a client as a file collection.

10       ContentLength

11       ToString

12  
13       [C#]           public           int           ContentLength           {get;}

14       [C++]          public:          \_\_property       int          get\_ContentLength();

15       [VB]   Public   ReadOnly   Property   ContentLength   As   Integer

16       [JScript]   public   function   get   ContentLength()   :   int;

17  
18       *Description*

19       Gets the size in bytes of an uploaded file.

20       ContentType

21       ToString

22  
23       [C#]           public           string           ContentType           {get;}

24       [C++]          public:          \_\_property       String\*       get\_ContentType();

25       [VB]   Public   ReadOnly   Property   ContentType   As   String



```
1 [JScript]      public      function      get      ContentType()      :      String;
```

```
2
```

```
3 Description
```

```
4      Gets the MIME content type of a file sent by a client.
```

```
5      FileName
```

```
6      ToString
```

```
7
```

```
8 [C#]           public           string           FileName           {get;}
```

```
9 [C++]          public:          __property          String*          get_FileName();
```

```
10 [VB]   Public   ReadOnly   Property   FileName   As   String
```

```
11 [JScript]      public      function      get      FileName()      :      String;
```

```
12
```

```
13 Description
```

```
14      Gets the fully-qualified name of the file on the client's machine (for
15 example "C:\MyFiles\Test.txt").
```

```
16      InputStream
```

```
17      ToString
```

```
18
```

```
19 [C#]           public           Stream           InputStream           {get;}
```

```
20 [C++]          public:          __property          Stream*          get_InputStream();
```

```
21 [VB]   Public   ReadOnly   Property   InputStream   As   Stream
```

```
22 [JScript]      public      function      get      InputStream()      :      Stream;
```

```
23
```

```
24 Description
```

```
25
```

1 Gets a **System.IO.Stream** object which points to an uploaded file to  
2 prepare for reading the contents of the file.

3 SaveAs

4  
5 [C#] public void SaveAs(string filename);

6 [C++] public: void SaveAs(String\* filename);

7 [VB] Public Sub SaveAs(ByVal filename As String)

8 [JScript] public function SaveAs(filename : String);

9  
10 *Description*

11 Saves an uploaded MIME message body to a file on the server. The name  
12 of the file.

13 HttpRequest class (System.Web)

14 ToString

15  
16  
17 *Description*

18 Enables ASP.NET to read the HTTP values sent by a client during a Web  
19 request.

20 HttpRequest

21 *Example Syntax:*

22 ToString

23  
24 [C#] public HttpRequest(string filename, string url, string queryString);

25 [C++] public: HttpRequest(String\* filename, String\* url, String\* queryString);

```

1 [VB] Public Sub New(ByVal filename As String, ByVal url As String, ByVal
2   queryString                               As                               String)

```

```

3 [JScript] public function HttpRequest(filename : String, url : String, queryString :
4   String);

```

#### 6 *Description*

7        Initializes an **System.Web.HttpRequest** object. The name of the file  
8 associated with the request. Information regarding the URL of the current request.  
9 The entire query string sent with the request (everything after the '?').

10        AcceptTypes

11        ToString

```

13 [C#]          public          string[]          AcceptTypes          {get;}

```

```

14 [C++]          public:          __property          String*          get_AcceptTypes();

```

```

15 [VB]   Public   ReadOnly   Property   AcceptTypes   As   String   ()

```

```

16 [JScript]   public   function   get   AcceptTypes()   :   String[];

```

#### 18 *Description*

19        Gets a string array of client-supported MIME accept types.

20        ApplicationPath

21        ToString

```

23 [C#]          public          string          ApplicationPath          {get;}

```

```

24 [C++]          public:          __property          String*          get_ApplicationPath();

```

```

25 [VB]   Public   ReadOnly   Property   ApplicationPath   As   String

```

1 [JScript] public function get ApplicationPath() : String;

2

3 *Description*

4 Gets the ASP.NET application's virtual application root path on the server.

5 Browser

6 ToString

7

8 [C#] public HttpBrowserCapabilities Browser {get; set;}

9 [C++] public: \_\_property HttpBrowserCapabilities\* get\_Browser();public:

10 \_\_property void set\_Browser(HttpBrowserCapabilities\*);

11 [VB] Public Property Browser As HttpBrowserCapabilities

12 [JScript] public function get Browser() : HttpBrowserCapabilities;public function

13 set Browser(HttpBrowserCapabilities);

14

15 *Description*

16 Gets information about the requesting client's browser capabilities.

17 ClientCertificate

18 ToString

19

20 [C#] public HttpClientCertificate ClientCertificate {get;}

21 [C++] public: \_\_property HttpClientCertificate\* get\_ClientCertificate();

22 [VB] Public ReadOnly Property ClientCertificate As HttpClientCertificate

23 [JScript] public function get ClientCertificate() : HttpClientCertificate;

24

25 *Description*

Gets the current request's client security certificate.

ContentEncoding

ToString

```
[C#]      public      Encoding      ContentEncoding      {get;}
```

```
[C++]     public:     __property     Encoding*      get_ContentEncoding();
```

```
[VB]     Public  ReadOnly  Property  ContentEncoding  As  Encoding
```

```
[JScript] public  function  get  ContentEncoding()  :  Encoding;
```

#### *Description*

Gets the character set of the entity-body.

Default **ContentEncoding** can be specified in an ASP.NET configuration file. If **ContentEncoding** is specified by the client, the default configuration settings are overridden.

ContentLength

ToString

```
[C#]      public      int      ContentLength      {get;}
```

```
[C++]     public:     __property     int      get_ContentLength();
```

```
[VB]     Public  ReadOnly  Property  ContentLength  As  Integer
```

```
[JScript] public  function  get  ContentLength()  :  int;
```

#### *Description*

Specifies the length, in bytes, of content sent by the client.

ContentType

ToString

```
[C#]          public          string          ContentType          {get;}
[C++]         public:         __property      String*          get_ContentType();
[VB]          Public          ReadOnly          Property      ContentType      As      String
[JScript]     public          function      get      ContentType()      :      String;
```

#### *Description*

Gets the MIME content type of the incoming request.

Cookies

ToString

```
[C#]          public          HttpCookieCollection      Cookies          {get;}
[C++]         public:         __property      HttpCookieCollection*      get_Cookies();
[VB]          Public          ReadOnly          Property      Cookies      As      HttpCookieCollection
[JScript]     public          function      get      Cookies()      :      HttpCookieCollection;
```

#### *Description*

Gets a collection of cookies sent by the client.

ASP.NET includes two intrinsic cookie collections. The collection accessed through **System.Web.HttpRequest.Cookies** contains cookies transmitted by the client to the server in the **Cookie** header. The collection accessed through **System.Web.HttpResponse.Cookies** contains cookies generated on the server and transmitted to the client in the **Set-Cookie** header.

CurrentExecutionFilePath

```

1      ToString
2
3  [C#]      public      string      CurrentExecutionFilePath      {get;}
4  [C++]     public:     __property  String*      get_CurrentExecutionFilePath();
5  [VB]      Public      ReadOnly  Property  CurrentExecutionFilePath  As  String
6  [JScript] public function get CurrentExecutionFilePath() : String;

```

```

7      FilePath

```

```

8      ToString

```

```

10 [C#]      public      string      FilePath      {get;}
11 [C++]     public:     __property  String*      get_FilePath();
12 [VB]      Public      ReadOnly  Property  FilePath  As  String
13 [JScript] public      function  get  FilePath()   :  String;

```

### 15 *Description*

16 Gets the virtual path of the current request.

17 The **System.Web.HttpRequest.FilePath** does not include the  
 18 **System.Web.HttpRequest.PathInfo** trailer. For the URL  
 19 **Http://www.microsoft.com/virdir/page.html/tail**, the **FilePath** is  
 20 **Http://www.microsoft.com/virdir/page.html** .

```

21      Files

```

```

22      ToString

```

```

24 [C#]      public      HttpFileCollection      Files      {get;}
25 [C++]     public:     __property  HttpFileCollection*      get_Files();

```

```

1  [VB]      Public      ReadOnly      Property      Files      As      HttpFileCollection
2  [JScript]  public      function      get      Files()      :      HttpFileCollection;

```

#### *Description*

Gets the collection of client-uploaded files (Multipart MIME format).

The file collection is populated only when the HTTP request Content-Type is multipart/form-data .

Filter

ToString

```

11 [C#]      public      Stream      Filter      {get;      set;}

```

```

12 [C++]  public:  __property  Stream*  get_Filter();public:  __property  void
13 set_Filter(Stream*);

```

```

14 [VB]      Public      Property      Filter      As      Stream

```

```

15 [JScript] public function get Filter() : Stream;public function set Filter(Stream);

```

#### *Description*

Gets or sets the filter to use when reading the current input stream.

Form

ToString

```

22 [C#]      public      NameValueCollection      Form      {get;}

```

```

23 [C++]  public:  __property  NameValueCollection*  get_Form();

```

```

24 [VB]      Public      ReadOnly      Property      Form      As      NameValueCollection

```

```

25 [JScript] public function get Form() : NameValueCollection;

```



1  
2 *Description*

3 Gets a collection of form variables.

4 Populated when the HTTP request Content-Type is either application/x-  
5 www-form-urlencoded or multipart/form-data .

6 Headers

7 ToString

8  
9 [C#] public NameValueCollection Headers {get;}

10 [C++] public: \_\_property NameValueCollection\* get\_Headers();

11 [VB] Public ReadOnly Property Headers As NameValueCollection

12 [JScript] public function get Headers() : NameValueCollection;

13  
14 *Description*

15 Gets a collection of HTTP headers.

16 HttpMethod

17 ToString

18  
19 [C#] public string HttpMethod {get;}

20 [C++] public: \_\_property String\* get\_HttpMethod();

21 [VB] Public ReadOnly Property HttpMethod As String

22 [JScript] public function get HttpMethod() : String;

23  
24 *Description*

1 Gets the HTTP data transfer method (such as **GET** , **POST** , or **HEAD** )  
2 used by the client.

3 **InputStream**

4 **ToString**

6 [C#] public Stream InputStream {get;}

7 [C++] public: \_\_property Stream\* get\_InputStream();

8 [VB] Public ReadOnly Property InputStream As Stream

9 [JScript] public function get InputStream() : Stream;

11 *Description*

12 Gets the contents of the incoming HTTP entity body.

13 **IsAuthenticated**

14 **ToString**

16 [C#] public bool IsAuthenticated {get;}

17 [C++] public: \_\_property bool get\_IsAuthenticated();

18 [VB] Public ReadOnly Property IsAuthenticated As Boolean

19 [JScript] public function get IsAuthenticated() : Boolean;

21 *Description*

22 Gets a value indicating whether the user has been authenticated.

23 **IsSecureConnection**

24 **ToString**

```

1
2 [C#]      public      bool      IsSecureConnection      {get;}
3 [C++]     public:     __property      bool      get_IsSecureConnection();
4 [VB]      Public      ReadOnly      Property      IsSecureConnection      As      Boolean
5 [JScript] public      function      get      IsSecureConnection()      :      Boolean;
6

```

#### 7 *Description*

8 Gets a value indicting whether the HTTP connection uses secure sockets  
9 (that is, HTTPS).

10 Item

11 ToString

```

12
13 [C#]      public      string      this[string      key]      {get;}
14 [C++]     public:     __property      String*      get_Item(String*      key);
15 [VB]      Public      Default      ReadOnly      Property      Item(ByVal key As String) As String
16 [JScript]      returnValue      =      HttpRequestObject.Item(key);
17

```

#### 18 *Description*

19 Default HttpRequest indexed property that retrieves a QueryString, Form,  
20 Cookies, or ServerVariables collection. This property is read-only. Numerical  
21 index to collection members.

22 Params

23 ToString

```

24
25 [C#]      public      NameValueCollection      Params      {get;}

```

```

1 [C++]      public:      __property      NameValueCollection*      get_Params();
2 [VB]      Public      ReadOnly      Property      Params      As      NameValueCollection
3 [JScript]      public      function      get      Params()      :      NameValueCollection;

```

#### *Description*

Gets a combined collection of **System.Web.HttpRequest.QueryString** , **System.Web.HttpRequest.Form** , **System.Web.HttpRequest.ServerVariables** , and **System.Web.HttpRequest.Cookies** items.

Path

ToString

```

12 [C#]          public          string          Path          {get;}
13 [C++]          public:          __property          String*          get_Path();
14 [VB]          Public          ReadOnly          Property          Path          As          String
15 [JScript]          public          function          get          Path()          :          String;

```

#### *Description*

Gets the virtual path of the current request.

The **System.Web.HttpRequest.FilePath** does not include the **System.Web.HttpRequest.PathInfo** trailer. For the URL **Http://www.microsoft.com/virdir/page.html/tail**, the **FilePath** is **Http://www.microsoft.com/virdir/page.html**.

PathInfo

ToString

```

1
2 [C#]          public          string          PathInfo          {get;}
3 [C++]         public:         __property      String*          get_PathInfo();
4 [VB]   Public   ReadOnly   Property   PathInfo   As   String
5 [JScript]    public   function   get   PathInfo()   :   String;
6

```

### *Description*

Gets additional path information for a resource with a URL extension.

For the URL [Http://www.microsoft.com/virdir/page.html/tail](http://www.microsoft.com/virdir/page.html/tail), the **PathInfo** value is /tail.

PhysicalApplicationPath

ToString

```

14 [C#]          public          string          PhysicalApplicationPath          {get;}
15 [C++]         public:         __property      String*          get_PhysicalApplicationPath();
16 [VB]   Public   ReadOnly   Property   PhysicalApplicationPath   As   String
17 [JScript]    public   function   get   PhysicalApplicationPath()   :   String;
18

```

### *Description*

Gets the physical file system path of the currently executing server application's root directory.

PhysicalPath

ToString

```

25 [C#]          public          string          PhysicalPath          {get;}

```

```

1  [C++]      public:      __property      String*      get_PhysicalPath();
2  [VB]      Public      ReadOnly      Property      PhysicalPath      As      String
3  [JScript]      public      function      get      PhysicalPath()      :      String;
4

```

#### *Description*

Gets the physical file system path corresponding to the requested URL.

QueryString

ToString

```

10 [C#]      public      NameValueCollection      QueryString      ,      {get;}
11 [C++]      public:      __property      NameValueCollection*      get_QueryString();
12 [VB]      Public      ReadOnly      Property      QueryString      As      NameValueCollection
13 [JScript]      public      function      get      QueryString()      :      NameValueCollection;
14

```

#### *Description*

Gets the collection of HTTP query string variables.

RawUrl

ToString

```

20 [C#]      public      string      RawUrl      {get;}
21 [C++]      public:      __property      String*      get_RawUrl();
22 [VB]      Public      ReadOnly      Property      RawUrl      As      String
23 [JScript]      public      function      get      RawUrl()      :      String;
24

```

#### *Description*

1 Gets the raw URL of the current request.

2 The raw URL is defined as the part of the URL following the domain  
3 information. In the URL string `http://www.microsoft.com/articles/recent.aspx`, the  
4 raw URL is `/articles/recent.aspx`. The raw URL includes the query string, if  
5 present.

6 RequestType

7 ToString

8  
9 [C#] public string RequestType {get; set;}

10 [C++] public: \_\_property String\* get\_RequestType();public: \_\_property void  
11 set\_RequestType(String\*);

12 [VB] Public Property RequestType As String

13 [JScript] public function get RequestType() : String;public function set  
14 RequestType(String);

15  
16 *Description*

17 Gets or sets the HTTP data transfer method ( **GET** or **POST** ) used by the  
18 client.

19 ServerVariables

20 ToString

21  
22 [C#] public NameValueCollection ServerVariables {get;}

23 [C++] public: \_\_property NameValueCollection\* get\_ServerVariables();

24 [VB] Public ReadOnly Property ServerVariables As NameValueCollection

25 [JScript] public function get ServerVariables() : NameValueCollection;

1  
2 *Description*

3 Gets a collection of web server variables.

4 TotalBytes

5 ToString

6  
7 [C#] public int TotalBytes {get;}

8 [C++] public: \_\_property int get\_TotalBytes();

9 [VB] Public ReadOnly Property TotalBytes As Integer

10 [JScript] public function get TotalBytes() : int;

11  
12 *Description*

13 Gets the number of bytes in the current input stream.

14 Url

15 ToString

16  
17 [C#] public Uri Url {get;}

18 [C++] public: \_\_property Uri\* get\_Url();

19 [VB] Public ReadOnly Property Url As Uri

20 [JScript] public function get Url() : Uri;

21  
22 *Description*

23 Gets Information about the URL of the current request.

24 UrlReferrer

25 ToString



```

1
2 [C#]          public          Uri          UrlReferrer          {get;}
3 [C++]         public:         __property    Uri*          get_UrlReferrer();
4 [VB]   Public  ReadOnly  Property  UrlReferrer  As  Uri
5 [JScript]    public    function    get    UrlReferrer()    :    Uri;
6

```

### *Description*

Gets information about the URL of the client's previous request that linked to the current URL.

UserAgent

ToString

```

13 [C#]          public          string          userAgent          {get;}
14 [C++]         public:         __property    String*          get_UserAgent();
15 [VB]   Public  ReadOnly  Property  userAgent  As  String
16 [JScript]    public    function    get    UserAgent()    :    String;
17

```

### *Description*

Gets the raw user agent string of the client browser.

UserHostAddress

ToString

```

23 [C#]          public          string          UserHostAddress          {get;}
24 [C++]         public:         __property    String*          get_UserHostAddress();
25 [VB]   Public  ReadOnly  Property  UserHostAddress  As  String

```

1 [JScript] public function get UserHostAddress() : String;

2

3 *Description*

4 Gets the IP host address of the remote client.

5 UserHostName

6 ToString

7

8 [C#] public string UserHostName {get;}

9 [C++] public: \_\_property String\* get\_UserHostName();

10 [VB] Public ReadOnly Property UserHostName As String

11 [JScript] public function get UserHostName() : String;

12

13 *Description*

14 Gets the DNS name of the remote client.

15 UserLanguages

16 ToString

17

18 [C#] public string[] UserLanguages {get;}

19 [C++] public: \_\_property String\* get\_UserLanguages();

20 [VB] Public ReadOnly Property UserLanguages As String ()

21 [JScript] public function get UserLanguages() : String[];

22

23 *Description*

24 Gets a sorted string array of client language preferences.

25 BinaryRead

```

1
2 [C#]      public      byte[]      BinaryRead(int      count);
3 [C++]    public:    unsigned    char    BinaryRead(int    count)    __gc[];
4 [VB]    Public Function BinaryRead(ByVal count As Integer) As Byte()
5 [JScript] public    function    BinaryRead(count    :    int)    :    Byte[];
6

```

### 7 *Description*

8 Performs a binary read of a specified number of bytes from the current  
9 input stream.

10 *Return Value:* A byte array.

11 The **BinaryRead** method is provided for compatibility with previous  
12 versions of ASP. Number of bytes to read.

### 13 **MapImageCoordinates**

```

14
15 [C#]    public    int[]    MapImageCoordinates(string    imageFieldName);
16 [C++]    public:    int    MapImageCoordinates(String*    imageFieldName)    __gc[];
17 [VB]    Public Function MapImageCoordinates(ByVal imageFieldName As String)
18 As                                             Integer()
19 [JScript] public function MapImageCoordinates(imageFieldName : String) : int[];
20

```

### 21 *Description*

22 Maps an incoming image-field form parameter to appropriate x/y  
23 coordinate values.

24 *Return Value:* A two-dimensional array of **integers** . A string reference to a form  
25 image map.

## MapPath

```
[C#]      public      string      MapPath(string      virtualPath);  
[C++]     public:     String*      MapPath(String*      virtualPath);  
[VB]      Public      Function      MapPath(ByVal virtualPath As String) As String  
[JScript] public function MapPath(virtualPath : String) : String; Maps the virtual  
path in the requested URL to a physical path on the server for the current request.
```

### *Description*

Maps the specified virtual path to a physical path. The virtual path (absolute or relative) for the current request.

## MapPath

```
[C#]      public      string      MapPath(string virtualPath, string baseVirtualDir, bool  
allowCrossAppMapping);  
[C++]     public:     String*      MapPath(String* virtualPath, String* baseVirtualDir, bool  
allowCrossAppMapping);  
[VB]      Public      Function      MapPath(ByVal virtualPath As String, ByVal  
baseVirtualDir As String, ByVal allowCrossAppMapping As Boolean) As String  
[JScript] public function MapPath(virtualPath : String, baseVirtualDir : String,  
allowCrossAppMapping      :      Boolean)      :      String;
```

### *Description*

1 Maps the specified virtual path to a physical path. The virtual path  
2 (absolute or relative) for the current request. The virtual base directory path used  
3 for relative resolution. If **true**, the *virtualPath* may belong to another application.

#### 4 SaveAs

6 [C#] public void SaveAs(string filename, bool includeHeaders);

7 [C++] public: void SaveAs(String\* filename, bool includeHeaders);

8 [VB] Public Sub SaveAs(ByVal filename As String, ByVal includeHeaders As  
9 Boolean)

10 [JScript] public function SaveAs(filename : String, includeHeaders : Boolean);

#### 12 *Description*

13 Saves an HTTP request to disk.

14 Saving the request context to disk can be useful in debugging. A string  
15 reference to a physical drive path. A **Boolean** value specifying whether an HTTP  
16 header should be saved to disk.

17 **HttpResponse** class (System.Web)

18 **ToString**

#### 21 *Description*

22 Encapsulates HTTP response information from an ASP.NET operation .

23 The methods and properties of the **HttpResponse** class are exposed  
24 through ASP.NET's intrinsic **Response** object.

25 **HttpResponse**

*Example Syntax:*

ToString

[C#]            public            HttpResponse(TextWriter            writer);

[C++]            public:            HttpResponse(TextWriter\*            writer);

[VB]    Public    Sub    New(ByVal    writer    As    TextWriter)

[JScript]    public    function    HttpResponse(writer    :    TextWriter);

*Description*

Initializes a new instance of the **HttpResponse** class. A **TextWriter** object enabling custom HTTP output.

Buffer

ToString

[C#]            public            bool            Buffer            {get;            set;}

[C++]    public:    \_\_property    bool    get\_Buffer();public:    \_\_property    void  
set\_Buffer(bool);

[VB]            Public            Property            Buffer            As            Boolean

[JScript]    public    function    get    Buffer()    :    Boolean;public    function    set  
Buffer(Boolean);

*Description*

Gets or sets a value indicating whether to buffer output and send it after the entire response is finished processing.

1       **System.Web.HttpResponse.Buffer** has been deprecated in favor of  
2       **System.Web.HttpResponse.BufferOutput** and is provided only for compatibility  
3       with previous versions of ASP. With ASP.NET, use  
4       **System.Web.HttpResponse.BufferOutput**.

5       BufferOutput

6       ToString

7  
8       [C#]       public       bool       BufferOutput       {get;       set;}

9       [C++]   public:   \_\_property bool   get\_BufferOutput();public:   \_\_property void  
10      set\_BufferOutput(bool);

11      [VB]       Public       Property       BufferOutput       As       Boolean

12      [JScript] public function get BufferOutput() : Boolean;public function set  
13      BufferOutput(Boolean);

14  
15      *Description*

16       Gets or sets a value indicating whether to buffer output and send it after the  
17      entire page is finished processing.

18       Cache

19       ToString

20  
21      [C#]       public       HttpCachePolicy       Cache       {get;}

22      [C++]   public:   \_\_property   HttpCachePolicy\*   get\_Cache();

23      [VB]   Public   ReadOnly   Property   Cache   As   HttpCachePolicy

24      [JScript] public function get Cache() : HttpCachePolicy;

1  
2 *Description*

3 Gets the caching policy (expiration time, privacy, vary clauses) of a Web  
4 page.

5 CacheControl

6 ToString

7  
8 [C#] public string CacheControl {get; set;}

9 [C++] public: \_\_property String\* get\_CacheControl();public: \_\_property void  
10 set\_CacheControl(String\*);

11 [VB] Public Property CacheControl As String

12 [JScript] public function get CacheControl() : String;public function set  
13 CacheControl(String);

14  
15 *Description*

16 Sets the **Cache-Control** HTTP header to **Public** or **Private** .

17 The values for **Private** and **Public** are strings and must be enclosed in  
18 quotation marks (" ").

19 Charset

20 ToString

21  
22 [C#] public string Charset {get; set;}

23 [C++] public: \_\_property String\* get\_Charset();public: \_\_property void  
24 set\_Charset(String\*);

25 [VB] Public Property Charset As String



1 [JScript] public function get Charset() : String;public function set Charset(String);

3 *Description*

4 Gets or sets the HTTP character set of the output stream.

5 *Charset* can be set to **null** to suppress the Content-Type header.

6 ContentEncoding

7 ToString

9 [C#] public Encoding ContentEncoding {get; set;}

10 [C++] public: \_\_property Encoding\* get\_ContentEncoding();public: \_\_property  
11 void set\_ContentEncoding(Encoding\*);

12 [VB] Public Property ContentEncoding As Encoding

13 [JScript] public function get ContentEncoding() : Encoding;public function set  
14 ContentEncoding(Encoding);

16 *Description*

17 Gets or sets the HTTP character set of the output stream.

18 ContentType

19 ToString

21 [C#] public string ContentType {get; set;}

22 [C++] public: \_\_property String\* get\_ContentType();public: \_\_property void  
23 set\_ContentType(String\*);

24 [VB] Public Property ContentType As String

25 [JScript] public function get ContentType() : String;public function set

1 ContentType(String);

2  
3 *Description*

4 Gets or sets the HTTP MIME type of the output stream.

5 The following example takes action if the content type of the output is not  
6 "Text/HTML".

7 Cookies

8 ToString

9  
10 [C#] public HttpCookieCollection Cookies {get;}

11 [C++] public: \_\_property HttpCookieCollection\* get\_Cookies();

12 [VB] Public ReadOnly Property Cookies As HttpCookieCollection

13 [JScript] public function get Cookies() : HttpCookieCollection;

14  
15 *Description*

16 Gets the response cookie collection.

17 ASP.NET includes two intrinsic cookie collections. The collection accessed  
18 through Cookies contains cookies transmitted by the client to the server in the  
19 **System.Web.HttpRequest.Cookies** header. The collection accessed through  
20 **System.Web.HttpResponse.Cookies** contains cookies generated on the server  
21 and transmitted to the client in the **Set-Cookie** header.

22 Expires

23 ToString

24  
25 [C#] public int Expires {get; set;}

```

1 [C++] public: __property int get_Expires();public: __property void
2 set_Expires(int);

```

```

3 [VB] Public Property Expires As Integer

```

```

4 [JScript] public function get Expires() : int;public function set Expires(int);

```

## Description

Gets or sets the number of minutes before a page cached on a browser expires. If the user returns to the same page before it expires, the cached version is displayed.

The **Expires** , **System.Web.HttpResponse.ExpiresAbsolute** and **System.Web.HttpResponse.CacheControl** properties have been deprecated in favor of the methods of the **System.Web.HttpCachePolicy** class available through the **System.Web.HttpResponse.Cache** intrinsic object to control the IIS output cache and client caches.

**ExpiresAbsolute**

**ToString**

```

18 [C#] public DateTime ExpiresAbsolute {get; set;}

```

```

19 [C++] public: __property DateTime get_ExpiresAbsolute();public: __property
20 void set_ExpiresAbsolute(DateTime);

```

```

21 [VB] Public Property ExpiresAbsolute As DateTime

```

```

22 [JScript] public function get ExpiresAbsolute() : DateTime;public function set
23 ExpiresAbsolute(DateTime);

```

## Description

1 Gets or sets the absolute date and time at which to remove cached  
2 information from the cache.

3 The **ExpiresAbsolute** , **System.Web.HttpResponse.Expires** and  
4 **System.Web.HttpResponse.CacheControl** properties have been deprecated in  
5 favor of the methods of the **System.Web.HttpCachePolicy** class available  
6 through the **System.Web.HttpResponse.Cache** intrinsic object to control the IIS  
7 output cache and client caches.

8 Filter

9 ToString

10  
11 [C#] public Stream Filter {get; set;}

12 [C++] public: \_\_property Stream\* get\_Filter();public: \_\_property void  
13 set\_Filter(Stream\*);

14 [VB] Public Property Filter As Stream

15 [JScript] public function get Filter() : Stream;public function set Filter(Stream);

16  
17 *Description*

18 Gets or sets a wrapping filter object used to modify the HTTP entity body  
19 before transmission.

20 When you create a **Stream** object and set the **Response.Filter** property to  
21 the **Stream** object, all HTTP output sent by **Response.Write** passes through the  
22 filter.

23 IsClientConnected

24 ToString

```

1
2 [C#]          public          bool          IsClientConnected          {get;}
3 [C++]         public:         __property     bool          get_IsClientConnected();
4 [VB]   Public  ReadOnly  Property  IsClientConnected  As  Boolean
5 [JScript]  public  function  get  IsClientConnected()  :  Boolean;
6

```

#### 7 *Description*

8 Gets a value indicating whether the client is still connected to the server.

9 Output

10 ToString

```

11
12 [C#]          public          TextWriter          Output          {get;}
13 [C++]         public:         __property     TextWriter*          get_Output();
14 [VB]   Public  ReadOnly  Property  Output  As  TextWriter
15 [JScript]  public  function  get  Output()  :  TextWriter;
16

```

#### 17 *Description*

18 Enables output of text to the outgoing HTTP response stream.

19 OutputStream

20 ToString

```

21
22 [C#]          public          Stream          OutputStream          {get;}
23 [C++]         public:         __property     Stream*          get_OutputStream();
24 [VB]   Public  ReadOnly  Property  OutputStream  As  Stream
25 [JScript]  public  function  get  OutputStream()  :  Stream;

```

## Description

Enables binary output to the outgoing HTTP content body.

Status

ToString

```
[C#]          public          string          Status          {get;          set;}
```

```
[C++] public: __property String* get_Status();public: __property void  
set_Status(String*);
```

```
[VB]          Public          Property          Status          As          String
```

```
[JScript] public function get Status() : String;public function set Status(String);
```

## Description

Sets the **Status** line that is returned to the client.

**System.Web.HttpResponse.Status** has been deprecated in favor of **System.Web.HttpResponse.StatusDescription** and is provided only for compatibility with previous versions of ASP. With ASP.NET, use **System.Web.HttpResponse.StatusDescription** instead.

StatusCode

ToString

```
[C#]          public          int          StatusCode          {get;          set;}
```

```
[C++] public: __property int get_StatusCode();public: __property void  
set_StatusCode(int);
```

```
[VB]          Public          Property          StatusCode          As          Integer
```

1 [JScript] public function get StatusCode() : int;public function set StatusCode(int);

2  
3 *Description*

4 Gets or sets the HTTP status code of the output returned to the client.

5 StatusDescription

6 ToString

7  
8 [C#] public string StatusDescription {get; set;}

9 [C++] public: \_\_property String\* get\_StatusDescription();public: \_\_property void  
10 set\_StatusDescription(String\*);

11 [VB] Public Property StatusDescription As String

12 [JScript] public function get StatusDescription() : String;public function set  
13 StatusDescription(String);

14  
15 *Description*

16 Gets or sets the HTTP status string of the output returned to the client.

17 SuppressContent

18 ToString

19  
20 [C#] public bool SuppressContent {get; set;}

21 [C++] public: \_\_property bool get\_SuppressContent();public: \_\_property void  
22 set\_SuppressContent(bool);

23 [VB] Public Property SuppressContent As Boolean

24 [JScript] public function get SuppressContent() : Boolean;public function set  
25 SuppressContent(Boolean);

1  
2 *Description*

3 Gets or sets a value indicating whether to send HTTP content to the client.

4 *AddCacheItemDependencies*

5  
6 [C#] public void AddCacheItemDependencies(ArrayList cacheKeys);

7 [C++] public: void AddCacheItemDependencies(ArrayList\* cacheKeys);

8 [VB] Public Sub AddCacheItemDependencies(ByVal cacheKeys As ArrayList)

9 [JScript] public function AddCacheItemDependencies(cacheKeys : ArrayList);

10 *AddCacheItemDependency*

11  
12 [C#] public void AddCacheItemDependency(string cacheKey);

13 [C++] public: void AddCacheItemDependency(String\* cacheKey);

14 [VB] Public Sub AddCacheItemDependency(ByVal cacheKey As String)

15 [JScript] public function AddCacheItemDependency(cacheKey : String);

16 *AddFileDependencies*

17  
18 [C#] public void AddFileDependencies(ArrayList filenames);

19 [C++] public: void AddFileDependencies(ArrayList\* filenames);

20 [VB] Public Sub AddFileDependencies(ByVal filenames As ArrayList)

21 [JScript] public function AddFileDependencies(filenames : ArrayList);

22  
23 *Description*

24 Adds a group of file names to the collection of file names on which the  
25 current response is dependent. The collection of files to add.



## AddFileDependency

```
[C#]      public      void      AddFileDependency(string      filename);
[C++]     public:     void      AddFileDependency(String*      filename);
[VB]     Public Sub   AddFileDependency(ByVal filename As String)
[JScript] public function AddFileDependency(filename : String);
```

### *Description*

Adds a single file name to the collection of file names on which the current response is dependent. The name of the file to add.

## AddHeader

```
[C#]      public      void      AddHeader(string      name,      string      value);
[C++]     public:     void      AddHeader(String*      name,      String*      value);
[VB]     Public Sub   AddHeader(ByVal name As String, ByVal value As String)
[JScript] public function AddHeader(name : String, value : String);
```

### *Description*

Adds an HTTP header to the output stream.

**AddHeader** is the same as

**System.Web.HttpResponse.AppendHeader(System.Web.HttpResponseHeader)** and is provided only for compatibility with previous versions of ASP. With ASP.NET, use **AppendHeader**. The name of the HTTP header to add *value* to. The string to add to the header.

## AppendCookie

```

1
2 [C#]      public      void      AppendCookie(HttpCookie      cookie);
3 [C++]     public:     void      AppendCookie(HttpCookie*      cookie);
4 [VB]      Public      Sub      AppendCookie(ByVal      cookie      As      HttpCookie)
5 [JScript] public      function AppendCookie(cookie      :      HttpCookie);
6

```

### *Description*

Adds an HTTP cookie to the intrinsic cookie collection. The cookie to add to the output stream.

### **AppendHeader**

```

11
12 [C#]      public      void      AppendHeader(string      name,      string      value);
13 [C++]     public:     void      AppendHeader(String*      name,      String*      value);
14 [VB]      Public      Sub      AppendHeader(ByVal      name      As      String, ByVal      value      As      String)
15 [JScript] public      function AppendHeader(name      :      String,      value      :      String);
16

```

### *Description*

Adds an HTTP header to the output stream.

If you use the

**System.Web.HttpResponse.AppendHeader(System.Web.HttpResponseHeader)** method to send cache-specific headers and at the same time use the cache object model ( **System.Web.HttpResponse.Cache** ) to set cache policy, HTTP response headers pertaining to caching ( **Cache-Control** , **Expires** , **Last-Modified** , **Pragma** , and **Vary**) might be deleted when the cache object model is used. This behavior enables ASP.NET to maintain the most restrictive settings. For example,

consider a page that includes user controls. If those controls have conflicting cache policies, the most restrictive cache policy will be used. If one user control sets the header " **Cache-Control: Public** " and another sets the more restrictive header " **Cache-Control: Private** " via calls to **System.Web.HttpCachePolicy.SetCacheability(System.Web.HttpCacheability )** , then the " **Cache-Control: Private** " header will be sent with the response. The name of the HTTP header to add to the output stream. The string to append to the header.

#### AppendToLog

```
[C#]      public      void      AppendToLog(string      param);
[C++]     public:     void      AppendToLog(String*      param);
[VB]      Public      Sub      AppendToLog(ByVal      param      As      String)
[JScript] public      function AppendToLog(param      :      String);
```

#### *Description*

Adds custom log information to the IIS log file. The text to add to the log file.

#### ApplyAppPathModifier

```
[C#]      public      string      ApplyAppPathModifier(string      virtualPath);
[C++]     public:     String*      ApplyAppPathModifier(String*      virtualPath);
[VB]      Public      Function      ApplyAppPathModifier(ByVal      virtualPath      As      String)      As      String
[JScript] public      function      ApplyAppPathModifier(virtualPath      :      String)      :      String;
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

*Description*

BinaryWrite

```
[C#]      public      void      BinaryWrite(byte[]      buffer);
[C++]     public:     void      BinaryWrite(unsigned      char      buffer      __gc[]);
[VB]      Public      Sub      BinaryWrite(ByVal      buffer()      As      Byte)
[JScript]      public      function      BinaryWrite(buffer      :      Byte[]);
```

*Description*

Writes a string of binary characters to the HTTP output stream. The bytes to write to the output stream.

Clear

```
[C#]      public      void      Clear();
[C++]     public:     void      Clear();
[VB]      Public      Sub      Clear()
[JScript]      public      function      Clear();
```

*Description*

Clears all content output from the buffer stream.

ClearContent

```
[C#]      public      void      ClearContent();
```

1	[C++]	public:	void	ClearContent();
2	[VB]	Public	Sub	ClearContent()
3	[JScript]	public	function	ClearContent();

4

5 *Description*

6       Clears all content output from the buffer stream.

7       ClearHeaders

8

9	[C#]	public	void	ClearHeaders();
10	[C++]	public:	void	ClearHeaders();
11	[VB]	Public	Sub	ClearHeaders()
12	[JScript]	public	function	ClearHeaders();

13

14 *Description*

15       Clears all headers from the buffer stream.

16       Close

17

18	[C#]	public	void	Close();
19	[C++]	public:	void	Close();
20	[VB]	Public	Sub	Close()
21	[JScript]	public	function	Close();

22

23 *Description*

24       Closes the socket connection to a client.

25       End

1				
2	[C#]	public	void	End();
3	[C++]	public:	void	End();
4	[VB]	Public	Sub	End()
5	[JScript]	public	function	End();

6

7 *Description*

8       Sends all currently buffered output to the client, stops execution of the

9 page, and fires the **Application\_EndRequest** event.

10       Flush

11				
12	[C#]	public	void	Flush();
13	[C++]	public:	void	Flush();
14	[VB]	Public	Sub	Flush()
15	[JScript]	public	function	Flush();

16

17 *Description*

18       Sends all currently buffered output to the client.

19       Forces all currently buffered output to be sent to the client.

20       Pics

21				
22	[C#]	public	void	Pics(string value);
23	[C++]	public:	void	Pics(String* value);
24	[VB]	Public	Sub	Pics(ByVal value As String)
25	[JScript]	public	function	Pics(value : String);

## Description

Appends a **PICS-Label** HTTP header to the output stream.

Platform for Internet Content Selection (PICS) is a World Wide Web Consortium (W3C) standard for content labeling. PICS is essentially a language for creating a ratings system. The string to add to the **PICS-Label** header.

### Redirect

```
[C#]      public      void      Redirect(string      url);
[C++]     public:     void      Redirect(String*      url);
[VB]      Public      Sub      Redirect(ByVal      url      As      String)
[JScript] public      function  Redirect(url      :      String);
```

## Description

Redirects a client to a new URL. The target location.

### Redirect

```
[C#]      public      void      Redirect(string      url,      bool      endResponse);
[C++]     public:     void      Redirect(String*      url,      bool      endResponse);
[VB]      Public      Sub      Redirect(ByVal url As String, ByVal endResponse As Boolean)
[JScript] public function Redirect(url : String, endResponse : Boolean); Redirects
a client to a new URL.
```

### RemoveOutputCacheItem

```
[C#]      public      static      void      RemoveOutputCacheItem(string      path);
```

```

1 [C++] public: static void RemoveOutputCacheItem(String* path);
2 [VB] Public Shared Sub RemoveOutputCacheItem(ByVal path As String)
3 [JScript] public static function RemoveOutputCacheItem(path : String);

```

#### SetCookie

```

6 [C#] public void SetCookie(HttpCookie cookie);
7 [C++] public: void SetCookie(HttpCookie* cookie);
8 [VB] Public Sub SetCookie(ByVal cookie As HttpCookie)
9 [JScript] public function SetCookie(cookie : HttpCookie);

```

#### Description

Updates an existing cookie in the cookie collection.

#### Write

```

15 [C#] public void Write(char ch);
16 [C++] public: void Write(__wchar_t ch);
17 [VB] Public Sub Write(ByVal ch As Char)
18 [JScript] public function Write(ch : Char);

```

#### Description

Writes a character to an HTTP output content stream. The character to write to the HTTP output stream.

#### Write

```

25 [C#] public void Write(object obj);

```



```

1  [C++]      public:      void      Write(Object*      obj);
2  [VB]      Public      Sub      Write(ByVal      obj      As      Object)
3  [JScript]      public      function      Write(obj      :      Object);
4

```

#### 5 *Description*

6 Writes an **Object** to an HTTP output content stream. The **Object** to write to  
7 the HTTP output stream.

#### 8 *Write*

```

9
10 [C#]      public      void      Write(string      s);
11 [C++]      public:      void      Write(String*      s);
12 [VB]      Public      Sub      Write(ByVal      s      As      String)
13 [JScript] public function Write(s : String); Writes information to an HTTP output
14 content      stream.
15

```

#### 16 *Description*

17 Writes a string to an HTTP output content stream. The string to write to the  
18 HTTP output stream.

#### 19 *Write*

```

20
21 [C#]      public      void      Write(char[]      buffer,      int      index,      int      count);
22 [C++]      public:      void      Write(__wchar_t      buffer      __gc[],      int      index,      int      count);
23 [VB]      Public      Sub      Write(ByVal      buffer()      As      Char,      ByVal      index      As      Integer,      ByVal
24 count      As      Integer)
25 [JScript] public function Write(buffer : Char[],`index : int, count : int);

```

## Description

Writes an array of characters to an HTTP output content stream. The character array to write. The position in the character array where writing starts. The number of characters to write, beginning at *index*.

### WriteFile

```
[C#]      public      void      WriteFile(string      filename);
[C++]      public:      void      WriteFile(String*      filename);
[VB]      Public      Sub      WriteFile(ByVal      filename      As      String)
[JScript] public function WriteFile(filename : String); Writes the specified file
directly      to      an      HTTP      content      output      stream.
```

## Description

Writes the specified file directly to an HTTP content output stream. The name of the file to write to the HTTP output.

### WriteFile

```
[C#]      public      void      WriteFile(string      filename,      bool      readIntoMemory);
[C++]      public:      void      WriteFile(String*      filename,      bool      readIntoMemory);
[VB]      Public      Sub      WriteFile(ByVal      filename      As      String,      ByVal      readIntoMemory      As
Boolean)
[JScript] public function WriteFile(filename : String, readIntoMemory : Boolean);
```

## Description

Writes the contents of the specified file into a memory block. The name of the file to write into a memory block. Indicates whether the file will be written into a memory block.

#### WriteFile

[C#] public void WriteFile(IntPtr fileHandle, long offset, long size);

[C++] public: void WriteFile(IntPtr fileHandle, \_\_int64 offset, \_\_int64 size);

[VB] Public Sub WriteFile(ByVal fileHandle As IntPtr, ByVal offset As Long, ByVal size As Long)

[JScript] public function WriteFile(fileHandle : IntPtr, offset : long, size : long);

#### *Description*

Writes the specified file directly to an HTTP content output stream. The file handle of the file to write to the HTTP output stream. The byte position in the file where writing will start. The number of bytes to write to the output stream.

#### WriteFile

[C#] public void WriteFile(string filename, long offset, long size);

[C++] public: void WriteFile(String\* filename, \_\_int64 offset, \_\_int64 size);

[VB] Public Sub WriteFile(ByVal filename As String, ByVal offset As Long, ByVal size As Long)

[JScript] public function WriteFile(filename : String, offset : long, size : long);

#### *Description*

1 Writes the specified file directly to an HTTP content output stream. The  
2 name of the file to write to the HTTP output stream. The byte position in the file  
3 where writing will start. The number of bytes to write to the output stream.

4 HttpRuntime class (System.Web)

5 WriteFile

6  
7  
8 *Description*

9 Provides a set of ASP.NET runtime services.

10 HttpRuntime

11 *Example Syntax:*

12 WriteFile

13  
14 [C#] public HttpRuntime();

15 [C++] public: HttpRuntime();

16 [VB] Public Sub New()

17 [JScript] public function HttpRuntime();

18 AppDomainAppId

19 WriteFile

20  
21 [C#] public static string AppDomainAppId {get;}

22 [C++] public: \_\_property static String\* get\_AppDomainAppId();

23 [VB] Public Shared ReadOnly Property AppDomainAppId As String

24 [JScript] public static function get AppDomainAppId() : String;

25

1  
2 *Description*

3  
4 AppDomainAppPath

5 WriteFile

6  
7 [C#] public static string AppDomainAppPath {get;}

8 [C++] public: \_\_property static String\* get\_AppDomainAppPath();

9 [VB] Public Shared ReadOnly Property AppDomainAppPath As String

10 [JScript] public static function get AppDomainAppPath() : String;

11  
12 *Description*

13  
14 AppDomainAppVirtualPath

15 WriteFile

16  
17 [C#] public static string AppDomainAppVirtualPath {get;}

18 [C++] public: \_\_property static String\* get\_AppDomainAppVirtualPath();

19 [VB] Public Shared ReadOnly Property AppDomainAppVirtualPath As String

20 [JScript] public static function get AppDomainAppVirtualPath() : String;

21  
22 *Description*

23  
24 AppDomainId

25 WriteFile

```

1
2 [C#]      public      static      string      AppDomainId      {get;}
3 [C++]     public:     __property static String*      get_AppDomainId();
4 [VB]      Public     Shared     ReadOnly     Property AppDomainId As String
5 [JScript] public     static     function     get      AppDomainId() : String;
6

```

### *Description*

```

9      AspInstallDirectory
10     WriteFile
11

```

```

12 [C#]      public      static      string      AspInstallDirectory      {get;}
13 [C++]     public:     __property static String*      get_AspInstallDirectory();
14 [VB]      Public     Shared     ReadOnly     Property AspInstallDirectory As String
15 [JScript] public     static     function     get      AspInstallDirectory() : String;
16

```

### *Description*

```

18
19     BinDirectory
20     WriteFile
21

```

```

22 [C#]      public      static      string      BinDirectory      {get;}
23 [C++]     public:     __property static String*      get_BinDirectory();
24 [VB]      Public     Shared     ReadOnly     Property BinDirectory As String
25 [JScript] public     static     function     get      BinDirectory() : String;

```

1  
2 *Description*

3  
4 Cache

5 WriteFile

6  
7 [C#] public static Cache Cache {get;}

8 [C++] public: \_\_property static Cache\* get\_Cache();

9 [VB] Public Shared ReadOnly Property Cache As Cache

10 [JScript] public static function get Cache() : Cache;

11  
12 *Description*

13 Provides access to the cache.

14 ClrInstallDirectory

15 WriteFile

16  
17 [C#] public static string ClrInstallDirectory {get;}

18 [C++] public: \_\_property static String\* get\_ClrInstallDirectory();

19 [VB] Public Shared ReadOnly Property ClrInstallDirectory As String

20 [JScript] public static function get ClrInstallDirectory() : String;

21  
22 *Description*

23  
24 CodegenDir

25 WriteFile

```

1
2 [C#]      public      static      string      CodegenDir      {get;}
3 [C++]     public:     __property static      String*      get_CodegenDir();
4 [VB]      Public      Shared      ReadOnly      Property      CodegenDir      As      String
5 [JScript] public      static      function      get      CodegenDir()      :      String;
6

```

### *Description*

IsOnUNCShare

WriteFile

```

12 [C#]      public      static      bool      IsOnUNCShare      {get;}
13 [C++]     public:     __property static      bool      get_IsOnUNCShare();
14 [VB]      Public      Shared      ReadOnly      Property      IsOnUNCShare      As      Boolean
15 [JScript] public      static      function      get      IsOnUNCShare()      :      Boolean;
16

```

### *Description*

MachineConfigurationDirectory

WriteFile

```

22 [C#]      public      static      string      MachineConfigurationDirectory      {get;}
23 [C++]     public:     __property static      String*      get_MachineConfigurationDirectory();
24 [VB]      Public      Shared      ReadOnly      Property      MachineConfigurationDirectory      As      String
25 [JScript] public      static      function      get      MachineConfigurationDirectory()      :      String;

```



1  
2 *Description*

3  
4 Close

5  
6 [C#] public static void Close();

7 [C++] public: static void Close();

8 [VB] Public Shared Sub Close()

9 [JScript] public static function Close();

10  
11 *Description*

12 Removes all items from the cache and shuts down the runtime.

13 ProcessRequest

14  
15 [C#] public static void ProcessRequest(HttpWorkerRequest wr);

16 [C++] public: static void ProcessRequest(HttpWorkerRequest\* wr);

17 [VB] Public Shared Sub ProcessRequest(ByVal wr As HttpWorkerRequest)

18 [JScript] public static function ProcessRequest(wr : HttpWorkerRequest);

19  
20 *Description*

21 The method that drives all ASP.NET Web processing execution.

22 HttpWorkerRequest object

23 HttpServerUtility class (System.Web)

24 ToString

### *Description*

Provides helper methods for processing Web requests.

The methods and properties of the **System.Web.HttpServerUtility** class are exposed through ASP.NET's intrinsic **System.Web.HttpContext.Server** object.

MachineName

ToString

[C#]            public            string            MachineName            {get;}

[C++]        public:        \_\_property        String\*        get\_MachineName();

[VB]        Public        ReadOnly        Property        MachineName        As        String

[JScript]    public        function        get        MachineName()        :        String;

### *Description*

Gets the server machine name.

ScriptTimeout

ToString

[C#]            public            int            ScriptTimeout            {get;        set;}

[C++]    public:    \_\_property    int    get\_ScriptTimeout();public:    \_\_property    void  
set\_ScriptTimeout(int);

[VB]        Public        Property        ScriptTimeout        As        Integer

[JScript]    public        function        get        ScriptTimeout()        :    int;public        function        set

1 ScriptTimeout(int);

2

3 *Description*

4 Gets and sets the request time-out in seconds.

5 ClearError

6

7 [C#] public void ClearError();

8 [C++] public: void ClearError();

9 [VB] Public Sub ClearError()

10 [JScript] public function ClearError();

11

12 *Description*

13 Clears the previous exception.

14 CreateObject

15

16 [C#] public object CreateObject(string progID);

17 [C++] public: Object\* CreateObject(String\* progID);

18 [VB] Public Function CreateObject(ByVal progID As String) As Object

19 [JScript] public function CreateObject(progID : String) : Object;

20

21 *Description*

22 Creates a server instance of a COM object identified by the object's

23 Programmatic Identifier (ProgID).

24 *Return Value:* The new object. The class or type of object to be instantiated.

25 CreateObject

```

1
2 [C#]      public      object      CreateObject(Type      type);
3 [C++]     public:     Object*      CreateObject(Type*      type);
4 [VB]      Public      Function      CreateObject(ByVal type As Type) As Object
5 [JScript] public      function      CreateObject(type : Type) : Object;
6

```

### *Description*

Instantiates a classic COM object identified via a Type.

### *CreateObjectFromClsid*

```

11 [C#]      public      object      CreateObjectFromClsid(string      clsid);
12 [C++]     public:     Object*      CreateObjectFromClsid(String*      clsid);
13 [VB]      Public      Function      CreateObjectFromClsid(ByVal clsid As String) As Object
14 [JScript] public      function      CreateObjectFromClsid(clsid : String) : Object;
15

```

### *Description*

Creates a server instance of a COM object identified by the object's class identifier (CLSID).

*Return Value:* The new object. The class identifier of the object to be instantiated.

### *Execute*

```

22 [C#]      public      void      Execute(string      path);
23 [C++]     public:     void      Execute(String*      path);
24 [VB]      Public      Sub      Execute(ByVal      path      As      String)
25 [JScript] public      function      Execute(path : String); Executes a request to another

```

1 page.

2  
3 *Description*

4 Executes a request to another page using the specified URL path to the  
5 page.

6 The **System.Web.HttpServerUtility.Execute(System.String)** method  
7 continues execution of the original page after execution of the new page is  
8 completed. The

9 **System.Web.HttpServerUtility.Transfer(System.String,System.Boolean)**  
10 method unconditionally transfers execution to another page. The URL path of the  
11 new request.

12 **Execute**

13  
14 [C#] public void Execute(string path, TextWriter writer);  
15 [C++] public: void Execute(String\* path, TextWriter\* writer);  
16 [VB] Public Sub Execute(ByVal path As String, ByVal writer As TextWriter)  
17 [JScript] public function Execute(path : String, writer : TextWriter);  
18

19 *Description*

20 Executes a request to another page using the specified URL path to the  
21 page. A **System.IO.TextWriter** captures output from the page.

22 The **System.Web.HttpServerUtility.Execute(System.String)** method  
23 continues execution of the original page after execution of the new page is  
24 completed. The

25 **System.Web.HttpServerUtility.Transfer(System.String,System.Boolean)**

method unconditionally transfers execution to another page. The URL path of the new request. The **System.IO.TextWriter** to capture the output.

### GetLastError

[C#]	public	Exception	GetLastError();
[C++]	public:	Exception*	GetLastError();
[VB]	Public	Function	GetLastError() As Exception
[JScript]	public	function	GetLastError() : Exception;

### Description

Returns the previous exception.

*Return Value:* The previous exception that was thrown.

### HtmlDecode

[C#]	public	string	HtmlDecode(string s);
[C++]	public:	String*	HtmlDecode(String* s);
[VB]	Public	Function	HtmlDecode(ByVal s As String) As String
[JScript]	public	function	HtmlDecode(s : String) : String;

Decodes a string that has been encoded to eliminate illegal HTML characters.

### Description

Decodes an HTML-encoded string and returns the decoded string.

*Return Value:* The decoded text.

URL encoding ensures that all browsers will correctly transmitted text in URL strings. Characters such as "?", "&", "/", and spaces may be truncated or

1 corrupted by some browsers so those characters cannot be used in  
2 ASP.NET pages in "" tags or in querystrings where the strings may be sent by a  
3 browser in a request string. The HTML string to decode.

#### 4 HtmlDecode

5  
6 [C#] public void HtmlDecode(string s, TextWriter output);  
7 [C++] public: void HtmlDecode(String\* s, TextWriter\* output);  
8 [VB] Public Sub HtmlDecode(ByVal s As String, ByVal output As TextWriter)  
9 [JScript] public function HtmlDecode(s : String, output : TextWriter);

#### 11 *Description*

12 Decodes an HTML-encoded string and sends the resulting output to a  
13 **System.IO.TextWriter** output stream.

14 URL encoding ensures that all browsers will correctly transmit text in URL  
15 strings. Characters such as "?", "&", "/", and spaces may be truncated or corrupted  
16 by some browsers so those characters cannot be used in ASP.NET pages in "" tags  
17 or in querystrings where the strings may be sent by a browser in a request string.  
18 The HTML string to decode. The **System.IO.TextWriter** output stream  
19 containing the decoded string.

#### 20 HtmlEncode

21  
22 [C#] public string HtmlEncode(string s);  
23 [C++] public: String\* HtmlEncode(String\* s);  
24 [VB] Public Function HtmlEncode(ByVal s As String) As String  
25 [JScript] public function HtmlEncode(s : String) : String; Encodes a string to be

1 displayed in a browser.

2  
3 *Description*

4 HTML-encodes a string and returns the encoded string.

5 *Return Value:* The HTML-encoded text.

6 URL encoding ensures that all browsers will correctly transmitted text in  
7 URL strings. Characters such as "?", "&", "/", and spaces may be truncated or  
8 corrupted by some browsers so those characters cannot be used in ASP.NET pages  
9 in "" tags or in querystrings where the strings may be sent by a browser in a  
10 request string. The text string to encode.

11 **HtmlEncode**

12  
13 [C#] public void HtmlEncode(string s, TextWriter output);

14 [C++] public: void HtmlEncode(String\* s, TextWriter\* output);

15 [VB] Public Sub HtmlEncode(ByVal s As String, ByVal output As TextWriter)

16 [JScript] public function HtmlEncode(s : String, output : TextWriter);

17  
18 *Description*

19 HTML-encodes a string and sends the resulting output to a  
20 **System.IO.TextWriter** output stream.

21 HTML encoding ensures that text will be correctly displayed in the  
22 browser, not interpreted by the browser as HTML. For example, if a text string  
23 contains "<" or ">" characters, the browser would interpret these characters as part  
24 of HTML tags. The HTML encoding of these two characters is "<" and ">",  
25 respectively, which causes the browser to display the angle brackets correctly. The



1 string to encode. The **System.IO.TextWriter** output stream containing the  
2 encoded string.

### 3 MapPath

4  
5 [C#] public string MapPath(string path);

6 [C++] public: String\* MapPath(String\* path);

7 [VB] Public Function MapPath(ByVal path As String) As String

8 [JScript] public function MapPath(path : String) : String;

### 9 10 Description

11 Returns the physical file path that corresponds to the specified virtual path  
12 on the Web server.

13 *Return Value:* The physical file path that corresponds to *path* . The virtual path on  
14 the Web server.

### 15 Transfer

16  
17 [C#] public void Transfer(string path);

18 [C++] public: void Transfer(String\* path);

19 [VB] Public Sub Transfer(ByVal path As String)

20 [JScript] public function Transfer(path : String);

### 21 22 Description

23 Terminates execution of the current page and begins execution of a new  
24 page using the specified URL path to the page. The URL path of the new page on  
25 the server to execute.

## Transfer

```
[C#]    public    void    Transfer(string    path,    bool    preserveForm);  
[C++]   public:   void    Transfer(String*    path,    bool    preserveForm);  
[VB]    Public Sub Transfer(ByVal path As String, ByVal preserveForm As  
Boolean)  
[JScript] public function Transfer(path : String, preserveForm : Boolean);
```

Terminates execution of the current page and begins execution of a new page.

### *Description*

Terminates execution of the current page and begins execution of a new page using the specified URL path to the page. Specifies whether to clear the **System.Web.HttpRequest.QueryString** and **System.Web.HttpRequest.Form** collections. The URL path of the new page on the server to execute. If **true**, the **QueryString** and **Form** collections are preserved. If **false**, they are cleared. The default is **false**.

## UrlDecode

```
[C#]          public          string          UrlDecode(string          s);  
[C++]         public:         String*         UrlDecode(String*         s);  
[VB]    Public Function UrlDecode(ByVal s As String) As String  
[JScript] public function UrlDecode(s : String) : String; Decodes a string encoded  
for HTTP transmission and sent to the server in a URL.
```

### *Description*

URL-decodes a string and returns the decoded string.

*Return Value:* The decoded text.

URL encoding ensures that all browsers will correctly transmitted text in URL strings. Characters such as "?", "&", "/", and spaces may be truncated or corrupted by some browsers so those characters cannot be used in ASP.NET pages in "" tags or in querystrings where the strings may be sent by a browser in a request string. The text string to decode.

#### UrlDecode

[C#] public void UrlDecode(string s, TextWriter output);

[C++] public: void UrlDecode(String\* s, TextWriter\* output);

[VB] Public Sub UrlDecode(ByVal s As String, ByVal output As TextWriter)

[JScript] public function UrlDecode(s : String, output : TextWriter);

#### Description

Decodes an HTML string received in a URL and sends the resulting output to a **System.IO.TextWriter** output stream.

URL encoding ensures that all browsers will correctly transmitted text in URL strings. Characters such as "?", "&", "/", and spaces may be truncated or corrupted by some browsers so those characters cannot be used in ASP.NET pages in "" tags or in querystrings where the strings may be sent by a browser in a request string. The HTML string to decode. The **System.IO.TextWriter** output stream containing the decoded string.

#### UrlEncode

```

1
2 [C#]      public      string      UriEncode(string      s);
3 [C++]     public:     String*      UriEncode(String*      s);
4 [VB]      Public      Function      UriEncode(ByVal s As String) As String
5 [JScript] public function UriEncode(s : String) : String; Encodes a string for
6 reliable HTTP transmission from the Web server to a client via the URL.

```

### 8 *Description*

9 URL-encodes a string and returns the encoded string.

10 *Return Value:* The URL encoded text.

11 URL encoding ensures that all browsers will correctly transmitted text in  
12 URL strings. Characters such as "?", "&", "/", and spaces may be truncated or  
13 corrupted by some browsers so those characters cannot be used in ASP.NET pages  
14 in "" tags or in querystrings where the strings may be sent by a browser in a  
15 request string. The text to URL-encode.

### 16 *UriEncode*

```

17
18 [C#]      public      void      UriEncode(string s, TextWriter output);
19 [C++]     public:     void      UriEncode(String* s, TextWriter* output);
20 [VB]      Public Sub UriEncode(ByVal s As String, ByVal output As TextWriter)
21 [JScript] public function UriEncode(s : String, output : TextWriter);

```

### 23 *Description*

24 URL encodes a string and sends the resulting output to a TextWriter output  
25 stream.

1 URL encoding ensures that all browsers will correctly transmitted text in  
2 URL strings. Characters such as "?", "&", "/", and spaces may be truncated or  
3 corrupted by some browsers so those characters cannot be used in ASP.NET pages  
4 in "" tags or in querystrings where the strings may be sent by a browser in a  
5 request string. The text string to encode. The **System.IO.TextWriter** output  
6 stream containing the encoded string.

### 7 UriPathEncode

8  
9 [C#] public string UriPathEncode(string s);  
10 [C++] public: String\* UriPathEncode(String\* s);  
11 [VB] Public Function UriPathEncode(ByVal s As String) As String  
12 [JScript] public function UriPathEncode(s : String) : String; Encodes the path  
13 portion of a URL string for reliable HTTP transmission from the Web server to a  
14 client via the URL.

### 16 Description

17 URL-encodes the path portion of a URL string and returns the encoded  
18 string.

19 *Return Value:* The URL encoded text.

20 URL encoding ensures that all browsers will correctly transmitted text in  
21 URL strings. Characters such as "?", "&", "/", and spaces may be truncated or  
22 corrupted by some browsers so those characters cannot cannot be used in  
23 ASP.NET pages in "" tags or in querystrings where the strings may be sent by a  
24 browser in a request string. The text to URL-encode.

25 HttpStaticObjectsCollection class (System.Web)

1        UrlPathEncode

2

3

4        *Description*

5        Provides        a        static        objects        collection        for        the  
6        **System.Web.HttpApplicationState.StaticObjects** property.

7        HttpStaticObjectsCollection

8        *Example Syntax:*

9        UrlPathEncode

10

11        [C#]                   public                   HttpStaticObjectsCollection();

12        [C++]                  public:                  HttpStaticObjectsCollection();

13        [VB]                  Public                   Sub                   New()

14        [JScript] public function HttpStaticObjectsCollection();

15        Count

16        UrlPathEncode

17

18        [C#]                  public                  int                  Count                  {get;}  
19        [C++]                  public:                  \_\_property                  int                  get\_Count();

20        [VB]        Public        ReadOnly        Property        Count        As        Integer

21        [JScript]        public        function        get        Count()        :        int;

22

23

24        *Description*

25        Gets the number of objects in the collection.

IsReadOnly

## UrlPathEncode

```
[C#]          public          bool          IsReadOnly          {get;}
[C++]         public:         __property     bool          get_IsReadOnly();
[VB]   Public   ReadOnly   Property   IsReadOnly   As   Boolean
[JScript]     public   function   get   IsReadOnly()   :   Boolean;
```

### *Description*

Gets a value indicating whether the collection is read-only.

IsSynchronized

UrlPathEncode

```
[C#]          public          bool          IsSynchronized          {get;}
[C++]         public:         __property     bool          get_IsSynchronized();
[VB]   Public   ReadOnly   Property   IsSynchronized   As   Boolean
[JScript]     public   function   get   IsSynchronized()   :   Boolean;
```

### *Description*

Gets a value indicating whether the collection is synchronized (i.e.: thread-safe).

Item

UrlPathEncode

```
[C#]          public          object          this[string          name]          {get;}
[C++]         public:         __property     Object*          get_Item(String*          name);
```

```

1 [VB] Public Default ReadOnly Property Item(ByVal name As String) As Object
2 [JScript]   returnValue   =   HttpStaticObjectsCollectionObject.Item(name);

```

#### *Description*

Gets the object with the specified name from the collection. The case-insensitive name of the object to get.

SyncRoot

UrlPathEncode

```

10 [C#]          public          object          SyncRoot          {get;}
11 [C++]          public:          __property          Object*          get_SyncRoot();
12 [VB]   Public   ReadOnly   Property   SyncRoot   As   Object
13 [JScript]   public   function   get   SyncRoot()   :   Object;

```

#### *Description*

Gets an object that can be used to synchronize access to the collection.

Program code should generally perform synchronized operations on the **SyncRoot** of a collection, not directly on the collection itself. This ensures proper operation of collections that are derived from other objects. Specifically, it maintains proper synchronization with other threads that might be simultaneously modifying the **collection** object.

CopyTo

```

24 [C#]          public          void          CopyTo(Array          array,          int          index);
25 [C++]          public:          __sealed          void          CopyTo(Array*          array,          int          index);

```



[VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As Integer)

[JScript] public function CopyTo(array : Array, index : int);

#### *Description*

Copies members of an **HttpStaticObjectsCollection** into an array. The array to copy the **HttpStaticObjectsCollection** into. The member of the collection where copying starts.

#### *GetEnumerator*

[C#] public IEnumerator GetEnumerator();

[C++] public: \_\_sealed IEnumerator\* GetEnumerator();

[VB] NotOverridable Public Function GetEnumerator() As IEnumerator

[JScript] public function GetEnumerator() : IEnumerator;

#### *Description*

Returns a dictionary enumerator used for iterating through the key-and-value pairs contained in the collection.

*Return Value:* The enumerator for the collection.

#### *GetObject*

[C#] public object GetObject(string name);

[C++] public: Object\* GetObject(String\* name);

[VB] Public Function GetObject(ByVal name As String) As Object

[JScript] public function GetObject(name : String) : Object;

1  
2 *Description*

3 Returns the object with the specified name from the collection. This  
4 property is an alternative to the **this** accessor.

5 *Return Value:* An object from the collection. The case-insensitive name of the  
6 object to return.

7 `HttpUnhandledException` class (System.Web)

8 `ToString`

9  
10  
11 *Description*

12 The exception that is thrown when a generic exception occurs.

13 `HttpUnhandledException`

14 *Example Syntax:*

15 `ToString`

16  
17 `[C#] public HttpUnhandledException(string message, Exception innerException);`

18 `[C++] public: HttpUnhandledException(String* message, Exception*`  
19 `innerException);`

20 `[VB] Public Sub New(ByVal message As String, ByVal innerException As`  
21 `Exception)`

22 `[JScript] public function HttpUnhandledException(message : String,`  
23 `innerException : Exception);` Initializes a new instance of the  
24 **System.Web.HttpUnhandledException** class.

## *Description*

Initializes a new instance of the **System.Web.HttpUnhandledException** class with the specified error message and inner exception. The message displayed to the client when the exception is thrown. The **System.Exception.InnerException**, if any, that threw the current exception.

**HttpUnhandledException**

*Example Syntax:*

**ToString**

```
[C#] public HttpUnhandledException(string message, string postMessage,
Exception innerException);
[C++] public: HttpUnhandledException(String* message, String* postMessage,
Exception* innerException);
[VB] Public Sub New(ByVal message As String, ByVal postMessage As String,
ByVal innerException As Exception)
[JScript] public function HttpUnhandledException(message : String, postMessage
: String, innerException : Exception);
```

## *Description*

Initializes a new instance of the **HttpUnhandledException** class with the specified error messages and inner exception. The message displayed to the client when the exception is thrown. An additional message displayed to the client when the exception is thrown. The **System.Exception.InnerException**, if any, that threw the current exception.

1	ErrorCode
2	HelpLink
3	HResult
4	InnerException
5	Message
6	Source
7	StackTrace
8	TargetSite
9	HttpUtility class (System.Web)
10	ToString

11

12

13 *Description*

14 Provides methods for encoding and decoding URLs when processing Web

15 requests.

16 HttpUtility

17 *Example Syntax:*

18 ToString

20	[C#]	public	HttpUtility();
21	[C++]	public:	HttpUtility();
22	[VB]	Public	Sub New()
23	[JScript] public function HttpUtility();		

24 HtmlAttributeEncode

25

```

1
2 [C#]      public      static      string      HtmlAttributeEncode(string      s);
3 [C++]     public:     static      String*      HtmlAttributeEncode(String*      s);
4 [VB] Public Shared Function HtmlAttributeEncode(ByVal s As String) As String
5 [JScript] public static function HtmlAttributeEncode(s : String) : String;
6 Minimally converts a string into an HTML-encoded string.

```

#### *Description*

Minimally converts a string to an HTML-encoded string.

*Return Value:* The encoded string.

**HtmlAttributeEncode** converts only quotes (") and ampersands (&) to equivalent character entities. It is considerably faster than the **System.Web.HttpUtility.HtmlEncode(System.String)** methods. The string to encode.

#### **HtmlAttributeEncode**

```

17 [C#] public static void HtmlAttributeEncode(string s, TextWriter output);
18 [C++] public: static void HtmlAttributeEncode(String* s, TextWriter* output);
19 [VB] Public Shared Sub HtmlAttributeEncode(ByVal s As String, ByVal output
20 As TextWriter)
21 [JScript] public static function HtmlAttributeEncode(s : String, output :
22 TextWriter);

```

#### *Description*

1 Minimally converts a string into an HTML-encoded string and sends the  
2 encoded string to a **System.IO.TextWriter** output stream.

3 **HtmlAttributeEncode** converts only quotes (") and ampersands (&) to  
4 equivalent character entities. It is considerably faster than the  
5 **System.Web.HttpUtility.HtmlEncode(System.String)** methods. The string to  
6 encode A **System.IO.TextWriter** output stream.

#### 7 HtmlDecode

8  
9 [C#] public static string HtmlDecode(string s);  
10 [C++] public: static String\* HtmlDecode(String\* s);  
11 [VB] Public Shared Function HtmlDecode(ByVal s As String) As String  
12 [JScript] public static function HtmlDecode(s : String) : String; Converts a string  
13 that has been HTML-encoded for HTTP transmission into a decoded string.

#### 15 *Description*

16 Converts a string that has been HTML-encoded for HTTP transmission into  
17 a decoded string.

18 *Return Value:* The decoded string.

19 If characters such as blanks and punctuation are passed in an HTTP stream,  
20 they might be misinterpreted at the receiving end. HTML encoding converts  
21 characters that are not allowed in HTML into character-entity equivalents; HTML  
22 decoding reverses the encoding. For example, when embedded in a block of text,  
23 the characters < and >, are encoded as < and > for HTTP transmission. The string  
24 to decode.

#### 25 HtmlDecode

```

1
2 [C#] public static void HtmlDecode(string s, TextWriter output);
3 [C++] public: static void HtmlDecode(String* s, TextWriter* output);
4 [VB] Public Shared Sub HtmlDecode(ByVal s As String, ByVal output As
5 TextWriter)
6 [JScript] public static function HtmlDecode(s : String, output : TextWriter);
7

```

### 8 *Description*

9 Converts a string that has been HTML-encoded into a decoded string, and  
10 sends the decoded string to a **System.IO.TextWriter** output stream.

11 If characters such as blanks and punctuation are passed in an HTTP stream,  
12 they might be misinterpreted at the receiving end. HTML encoding converts  
13 characters that are not allowed in HTML into character-entity equivalents; HTML  
14 decoding reverses the encoding. For example, when embedded in a block of text,  
15 the characters < and >, are encoded as < and > for HTTP transmission. The string  
16 to decode. A **System.IO.TextWriter** stream of output.

### 17 **HtmlEncode**

```

18
19 [C#] public static string HtmlEncode(string s);
20 [C++] public: static String* HtmlEncode(String* s);
21 [VB] Public Shared Function HtmlEncode(ByVal s As String) As String
22 [JScript] public static function HtmlEncode(s : String) : String; Converts a string
23 into an HTML-encoded string for reliable HTTP transmission from the Web
24 server to a client.
25

```

## Description

Converts a string to an HTML-encoded string for reliable HTTP transmission from the Web server to a client.

*Return Value:* The encoded string.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. HTML encoding converts characters that are not allowed in HTML into character-entity equivalents; HTML decoding reverses the encoding. For example, when embedded in a block of text, the characters < and >, are encoded as < and > for HTTP transmission. The string to encode.

## HtmlEncode

```
[C#] public static void HtmlEncode(string s, TextWriter output);  
[C++] public: static void HtmlEncode(String* s, TextWriter* output);  
[VB] Public Shared Sub HtmlEncode(ByVal s As String, ByVal output As  
TextWriter)  
[JScript] public static function HtmlEncode(s : String, output : TextWriter);
```

## Description

Converts a string into an HTML-encoded string, and returns the output as a TextWriter stream of output.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. HTML encoding converts characters that are not allowed in HTML into character-entity equivalents; HTML





```
1 [VB] Public Shared Function UriDecode(ByVal bytes() As Byte, ByVal e As  
2 Encoding) As String
```

```
3 [JScript] public static function UriDecode(bytes : Byte[], e : Encoding) : String;
```

#### 4 5 *Description*

6 Converts a URL-encoded byte array into a decoded string, using the  
7 specified decoding method.

8 *Return Value:* The decoded string.

9 If characters such as blanks and punctuation are passed in an HTTP stream,  
10 they might be misinterpreted at the receiving end. URL encoding converts  
11 characters that are not allowed in a URL into character-entity equivalents; URL  
12 decoding reverses the encoding. For example, when embedded in a block of text to  
13 be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The  
14 byte array to decode. The **System.Text.Encoding** that specifies the decoding  
15 method.

#### 16 UriDecode

```
17  
18 [C#] public static string UriDecode(string str, Encoding e);
```

```
19 [C++] public: static String* UriDecode(String* str, Encoding* e);
```

```
20 [VB] Public Shared Function UriDecode(ByVal str As String, ByVal e As  
21 Encoding) As String
```

```
22 [JScript] public static function UriDecode(str : String, e : Encoding) : String;
```

#### 23 24 *Description*

25

Converts a URL-encoded string into a decoded string, using the specified decoding method.

*Return Value:* The decoded string.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The string to decode. The **System.Text.Encoding** that specifies the decoding method.

#### UrlDecode

```
[C#] public static string UrlDecode(byte[] bytes, int offset, int count, Encoding e);  
[C++] public: static String* UrlDecode(unsigned char bytes __gc[], int offset, int  
count, Encoding* e);  
[VB] Public Shared Function UrlDecode(ByVal bytes() As Byte, ByVal offset As  
Integer, ByVal count As Integer, ByVal e As Encoding) As String  
[JScript] public static function UrlDecode(bytes : Byte[], offset : int, count : int, e :  
Encoding) : String;
```

#### Description

Converts a URL-encoded byte array into a decoded string, using the specified decoding method, starting at the specified position in the array, and continuing for the specified number of bytes.

*Return Value:* The decoded string.

1 If characters such as blanks and punctuation are passed in an HTTP stream,  
2 they might be misinterpreted at the receiving end. URL encoding converts  
3 characters that are not allowed in a URL into character-entity equivalents; URL  
4 decoding reverses the encoding. For example, when embedded in a block of text to  
5 be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The  
6 array of bytes to decode. The position in the byte to begin decoding. The number  
7 of bytes to decode starting at *offset*. The **System.Text.Encoding** object that  
8 specifies the decoding method.

### 9 UriDecodeToBytes

10  
11 [C#] public static byte[] UriDecodeToBytes(byte[] bytes);  
12 [C++] public: static unsigned char UriDecodeToBytes(unsigned char bytes \_\_gc[])  
13 \_\_gc[];  
14 [VB] Public Shared Function UriDecodeToBytes(ByVal bytes() As Byte) As  
15 Byte()  
16 [JScript] public static function UriDecodeToBytes(bytes : Byte[]) : Byte[];

### 18 Description

19 Converts a URL-encoded array of bytes into a decoded array of bytes.

20 *Return Value:* The decoded array of bytes.

21 If characters such as blanks and punctuation are passed in an HTTP stream,  
22 they might be misinterpreted at the receiving end. URL encoding converts  
23 characters that are not allowed in a URL into character-entity equivalents; URL  
24 decoding reverses the encoding. For example, when embedded in a block of text to  
25

1 be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The  
2 array of bytes to decode.

### 3       UrlDecodeToBytes

4  
5 [C#]       public       static       byte[]       UrlDecodeToBytes(string       str);  
6 [C++]       public: static unsigned char UrlDecodeToBytes(String\* str) \_\_gc[];  
7 [VB] Public Shared Function UrlDecodeToBytes(ByVal str As String) As Byte()  
8 [JScript] public static function UrlDecodeToBytes(str : String) : Byte[]; Converts a  
9 URL-encoded string or byte array into a decoded array of bytes.

### 11      *Description*

12       Converts a URL-encoded string into a decoded array of bytes .

13      *Return Value:* The decoded array of bytes.

14       If characters such as blanks and punctuation are passed in an HTTP stream,  
15 they might be misinterpreted at the receiving end. URL encoding converts  
16 characters that are not allowed in a URL into character-entity equivalents; URL  
17 decoding reverses the encoding. For example, when embedded in a block of text to  
18 be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The  
19 string to decode.

### 20       UrlDecodeToBytes

21  
22 [C#]       public       static       byte[]       UrlDecodeToBytes(string       str,       Encoding       e);  
23 [C++]       public: static unsigned char UrlDecodeToBytes(String\* str, Encoding\* e)  
24       \_\_gc[];  
25 [VB] Public Shared Function UrlDecodeToBytes(ByVal str As String, ByVal e As

```

1 Encoding)                                     As                                     Byte()
2 [JScript] public static function UriDecodeToBytes(str : String, e : Encoding) :
3 Byte[];

```

#### 5 *Description*

6 Converts a URL-encoded string into a decoded array of bytes, using the  
7 specified decoding method.

8 *Return Value:* The decoded array of bytes.

9 If characters such as blanks and punctuation are passed in an HTTP stream,  
10 they might be misinterpreted at the receiving end. URL encoding converts  
11 characters that are not allowed in a URL into character-entity equivalents; URL  
12 decoding reverses the encoding. For example, when embedded in a block of text to  
13 be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The  
14 string to decode. The **System.Text.Encoding** object that specifies the decoding  
15 method.

#### 16 UriDecodeToBytes

```

17
18 [C#] public static byte[] UriDecodeToBytes(byte[] bytes, int offset, int count);
19 [C++] public: static unsigned char UriDecodeToBytes(unsigned char bytes __gc[],
20 int offset, int count) __gc[];
21 [VB] Public Shared Function UriDecodeToBytes(ByVal bytes() As Byte, ByVal
22 offset As Integer, ByVal count As Integer) As Byte()
23 [JScript] public static function UriDecodeToBytes(bytes : Byte[], offset : int,
24 count : int) : Byte[];

```

## Description

Converts a URL-encoded array of bytes into a decoded array of bytes, starting at the specified position in the array and continuing for the specified number of bytes.

*Return Value:* The decoded array of bytes.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The array of bytes to decode. The position in the byte array to begin decoding. The number of bytes to decode, starting at *offset*.

## UrlEncode

```
[C#]      public      static      string      UrlEncode(byte[]      bytes);  
[C++]    public:  static  String*  UrlEncode(unsigned char bytes __gc[]);  
[VB]     Public Shared Function UrlEncode(ByVal bytes() As Byte) As String  
[JScript] public static function UrlEncode(bytes : Byte[]) : String;
```

## Description

Converts a byte array into an encoded URL string for reliable HTTP transmission from the Web server to a client.

*Return Value:* The encoded string.

1 If characters such as blanks and punctuation are passed in an HTTP stream,  
2 they might be misinterpreted at the receiving end. URL encoding converts  
3 characters that are not allowed in a URL into character-entity equivalents; URL  
4 decoding reverses the encoding. For example, when embedded in a block of text to  
5 be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The  
6 byte array to encode.

## 7 UriEncode

9 [C#] public static string UriEncode(string str);  
10 [C++] public: static String\* UriEncode(String\* str);  
11 [VB] Public Shared Function UriEncode(ByVal str As String) As String  
12 [JScript] public static function UriEncode(str : String) : String; Encodes a URL  
13 string for reliable HTTP transmission from the Web server to a client.

## 15 Description

16 Encodes a URL string for reliable HTTP transmission from the Web server  
17 to a client.

18 *Return Value:* The encoded string.

19 If characters such as blanks and punctuation are passed in an HTTP stream,  
20 they might be misinterpreted at the receiving end. URL encoding converts  
21 characters that are not allowed in a URL into character-entity equivalents; URL  
22 decoding reverses the encoding. For example, when embedded in a block of text to  
23 be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The  
24 text to encode.

## 25 UriEncode



```

1
2 [C#] public static string UriEncode(string str, Encoding e);
3 [C++] public: static String* UriEncode(String* str, Encoding* e);
4 [VB] Public Shared Function UriEncode(ByVal str As String, ByVal e As
5 Encoding) As String
6 [JScript] public static function UriEncode(str : String, e : Encoding) : String;
7

```

### 8 *Description*

9 Encodes a URL string for reliable HTTP transmission from the Web server  
10 to a client, using the specified encoding method.

11 *Return Value:* The encoded string.

12 If characters such as blanks and punctuation are passed in an HTTP stream,  
13 they might be misinterpreted at the receiving end. URL encoding converts  
14 characters that are not allowed in a URL into character-entity equivalents; URL  
15 decoding reverses the encoding. For example, when embedded in a block of text to  
16 be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The  
17 text to encode. The **System.Text.Encoding** object that specifies the encoding  
18 scheme.

### 19 UriEncode

```

20
21 [C#] public static string UriEncode(byte[] bytes, int offset, int count);
22 [C++] public: static String* UriEncode(unsigned char bytes __gc[], int offset, int
23 count);
24 [VB] Public Shared Function UriEncode(ByVal bytes() As Byte, ByVal offset As
25 Integer, ByVal count As Integer) As String

```

```
1 [JScript] public static function UrlEncode(bytes : Byte[], offset : int, count : int) :  
2 String;
```

#### 4 *Description*

5 Converts a byte array into a URL-encoded string for reliable HTTP  
6 transmission from the Web server to a client, starting at the specified position in  
7 the array and continuing for the specified number of bytes.

8 *Return Value:* The encoded string.

9 If characters such as blanks and punctuation are passed in an HTTP stream,  
10 they might be misinterpreted at the receiving end. URL encoding converts  
11 characters that are not allowed in a URL into character-entity equivalents; URL  
12 decoding reverses the encoding. For example, when embedded in a block of text to  
13 be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The  
14 byte array to encode. The position in the byte array to begin encoding. The number  
15 of bytes to encode, starting at *offset*.

#### 16 *UrlEncodeToBytes*

```
18 [C#] public static byte[] UrlEncodeToBytes(byte[] bytes);
```

```
19 [C++] public: static unsigned char UrlEncodeToBytes(unsigned char bytes __gc[])  
20 __gc[];
```

```
21 [VB] Public Shared Function UrlEncodeToBytes(ByVal bytes() As Byte) As  
22 Byte()
```

```
23 [JScript] public static function UrlEncodeToBytes(bytes : Byte[]) : Byte[];
```

#### 25 *Description*

Converts an array of bytes into a URL-encoded array of bytes.

*Return Value:* The encoded array of bytes.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The array of bytes to encode.

#### UrlEncodeToBytes

[C#] public static byte[] UrlEncodeToBytes(string str);

[C++] public: static unsigned char UrlEncodeToBytes(String\* str) \_\_gc[];

[VB] Public Shared Function UrlEncodeToBytes(ByVal str As String) As Byte()

[JScript] public static function UrlEncodeToBytes(str : String) : Byte[]; Converts a string or a byte array into an encoded array of bytes for reliable HTTP transmission from the Web server to a client.

#### Description

Converts a string into a URL-encoded array of bytes .

*Return Value:* The encoded array of bytes.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to

1 be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The  
2 string to encode.

### 3 UriEncodeToBytes

4  
5 [C#] public static byte[] UriEncodeToBytes(string str, Encoding e);

6 [C++] public: static unsigned char UriEncodeToBytes(String\* str, Encoding\* e)  
7 \_\_gc[];

8 [VB] Public Shared Function UriEncodeToBytes(ByVal str As String, ByVal e As  
9 Encoding) As Byte()

10 [JScript] public static function UriEncodeToBytes(str : String, e : Encoding) :  
11 Byte[];

### 12 13 *Description*

14 Converts a string into a URL-encoded array of bytes, using the specified  
15 encoding method.

16 *Return Value:* The encoded array of bytes.

17 If characters such as blanks and punctuation are passed in an HTTP stream,  
18 they might be misinterpreted at the receiving end. URL encoding converts  
19 characters that are not allowed in a URL into character-entity equivalents; URL  
20 decoding reverses the encoding. For example, when embedded in a block of text to  
21 be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The  
22 string to encode The **System.Text.Encoding** that specifies the encoding method.

### 23 UriEncodeToBytes

24  
25 [C#] public static byte[] UriEncodeToBytes(byte[] bytes, int offset, int count);

```

1 [C++] public: static unsigned char UrlEncodeToBytes(unsigned char bytes __gc[],
2 int offset, int count) __gc[];
3 [VB] Public Shared Function UrlEncodeToBytes(ByVal bytes() As Byte, ByVal
4 offset As Integer, ByVal count As Integer) As Byte()
5 [JScript] public static function UrlEncodeToBytes(bytes : Byte[], offset : int, count
6 : int) : Byte[];

```

### Description

Converts an array of bytes into a URL-encoded array of bytes, starting at the specified position in the array and continuing for the specified number of bytes.

*Return Value:* The encoded array of bytes.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The array of bytes to encode. The position in the byte array to begin encoding. The number of bytes to encode, starting at *offset*.

### UrlEncodeUnicode

```

22 [C#] public static string UrlEncodeUnicode(string str);
23 [C++] public: static String* UrlEncodeUnicode(String* str);
24 [VB] Public Shared Function UrlEncodeUnicode(ByVal str As String) As String
25 [JScript] public static function UrlEncodeUnicode(str : String) : String;

```

1  
2 *Description*

3 Converts a string into a Unicode string.

4 *Return Value:* The Unicode string.

5 If characters such as blanks and punctuation are passed in an HTTP stream,  
6 they might be misinterpreted at the receiving end. URL encoding converts  
7 characters that are not legal in a URL to character-entity equivalents; URL  
8 decoding reverses the encoding. For example, when embedded in a block of text to  
9 be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The  
10 string to convert.

11 `UrlEncodeUnicodeToBytes`

12  
13 [C#] public static byte[] UrlEncodeUnicodeToBytes(string str);

14 [C++] public: static unsigned char UrlEncodeUnicodeToBytes(String\* str) \_\_gc[];

15 [VB] Public Shared Function UrlEncodeUnicodeToBytes(ByVal str As String) As  
16 Byte()

17 [JScript] public static function UrlEncodeUnicodeToBytes(str : String) : Byte[];

18  
19 *Description*

20 Converts a string into a Unicode array of bytes.

21 *Return Value:* The Unicode byte array.

22 If characters such as blanks and punctuation are passed in an HTTP stream,  
23 they might be misinterpreted at the receiving end. URL encoding converts  
24 characters that are not legal in a URL to character-entity equivalents; URL  
25 decoding reverses the encoding. For example, when embedded in a block of text to

1 be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The  
2 string to convert.

3       HttpValidationStatus enumeration (System.Web)

4       UrlEncodeUnicodeToBytes

7       *Description*

8       Provides enumerated values that indicate cache validation status.

9       UrlEncodeUnicodeToBytes

11      [C#]       public       const       HttpValidationStatus       IgnoreThisRequest;

12      [C++]       public:       const       HttpValidationStatus       IgnoreThisRequest;

13      [VB]       Public       Const       IgnoreThisRequest       As       HttpValidationStatus

14      [JScript]       public       var       IgnoreThisRequest       :       HttpValidationStatus;

16       *Description*

17       Specifies not to validate the entity in the cache.

18       UrlEncodeUnicodeToBytes

20      [C#]       public       const       HttpValidationStatus       Invalid;

21      [C++]       public:       const       HttpValidationStatus       Invalid;

22      [VB]       Public       Const       Invalid       As       HttpValidationStatus

23      [JScript]       public       var       Invalid       :       HttpValidationStatus;

25       *Description*

1 Indicates that the cache is invalid.

2 UriEncodeUnicodeToBytes

3

4 [C#] public const HttpValidationStatus Valid;

5 [C++] public: const HttpValidationStatus Valid;

6 [VB] Public Const Valid As HttpValidationStatus

7 [JScript] public var Valid : HttpValidationStatus;

8

9 *Description*

10 Indicates that the cache is valid.

11 HttpWorkerRequest class (System.Web)

12 ToString

13

14

15 *Description*

16 This abstract class defines the base worker methods and enumerations used

17 by ASP.NET managed code for request processing.

18 ToString

19

20 [C#] public const int HeaderAccept;

21 [C++] public: const int HeaderAccept;

22 [VB] Public Const HeaderAccept As Integer

23 [JScript] public var HeaderAccept : int;

24

25 *Description*



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

ToString

[C#]	public	const	int	HeaderAcceptCharset;
[C++]	public:	const	int	HeaderAcceptCharset;
[VB]	Public	Const	HeaderAcceptCharset	As Integer
[JScript]	public	var	HeaderAcceptCharset	: int;

Description

ToString

[C#]	public	const	int	HeaderAcceptEncoding;
[C++]	public:	const	int	HeaderAcceptEncoding;
[VB]	Public	Const	HeaderAcceptEncoding	As Integer
[JScript]	public	var	HeaderAcceptEncoding	: int;

Description

ToString

[C#]	public	const	int	HeaderAcceptLanguage;
[C++]	public:	const	int	HeaderAcceptLanguage;
[VB]	Public	Const	HeaderAcceptLanguage	As Integer
[JScript]	public	var	HeaderAcceptLanguage	: int;

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

*Description*

ToString

```
[C#]      public      const      int      HeaderAcceptRanges;  
[C++]     public:     const      int      HeaderAcceptRanges;  
[VB]      Public      Const      HeaderAcceptRanges      As      Integer  
[JScript] public      var      HeaderAcceptRanges      :      int;
```

*Description*

ToString

```
[C#]      public      const      int      HeaderAge;  
[C++]     public:     const      int      HeaderAge;  
[VB]      Public      Const      HeaderAge      As      Integer  
[JScript] public      var      HeaderAge      :      int;
```

*Description*

ToString

```
[C#]      public      const      int      HeaderAllow;  
[C++]     public:     const      int      HeaderAllow;
```

1	[VB]	Public	Const	HeaderAllow	As	Integer
2	[JScript]	public	var	HeaderAllow	:	int;

3

4 *Description*

5

6 ToString

7

8 [C#] public const int HeaderAuthorization;

9 [C++] public: const int HeaderAuthorization;

10 [VB] Public Const HeaderAuthorization As Integer

11 [JScript] public var HeaderAuthorization : int;

12

13 *Description*

14

15 ToString

16

17 [C#] public const int HeaderCacheControl;

18 [C++] public: const int HeaderCacheControl;

19 [VB] Public Const HeaderCacheControl As Integer

20 [JScript] public var HeaderCacheControl : int;

21

22 *Description*

23

24 ToString

25

1  
2 [C#] public const int HeaderConnection;  
3 [C++] public: const int HeaderConnection;  
4 [VB] Public Const HeaderConnection As Integer  
5 [JScript] public var HeaderConnection : int;

6  
7 *Description*

8  
9 ToString

10  
11 [C#] public const int HeaderContentEncoding;  
12 [C++] public: const int HeaderContentEncoding;  
13 [VB] Public Const HeaderContentEncoding As Integer  
14 [JScript] public var HeaderContentEncoding : int;

15  
16 *Description*

17  
18 ToString

19  
20 [C#] public const int HeaderContentLanguage;  
21 [C++] public: const int HeaderContentLanguage;  
22 [VB] Public Const HeaderContentLanguage As Integer  
23 [JScript] public var HeaderContentLanguage : int;

24  
25 *Description*

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

ToString

[C#] public const int HeaderContentLength;  
[C++] public: const int HeaderContentLength;  
[VB] Public Const HeaderContentLength As Integer  
[JScript] public var HeaderContentLength : int;

Description

ToString

[C#] public const int HeaderContentLocation;  
[C++] public: const int HeaderContentLocation;  
[VB] Public Const HeaderContentLocation As Integer  
[JScript] public var HeaderContentLocation : int;

Description

ToString

[C#] public const int HeaderContentMd5;  
[C++] public: const int HeaderContentMd5;  
[VB] Public Const HeaderContentMd5 As Integer  
[JScript] public var HeaderContentMd5 : int;

1  
2 *Description*

3  
4 ToString

5  
6 [C#] public const int HeaderContentRange;

7 [C++] public: const int HeaderContentRange;

8 [VB] Public Const HeaderContentRange As Integer

9 [JScript] public var HeaderContentRange : int;

10  
11 *Description*

12  
13 ToString

14  
15 [C#] public const int HeaderContentType;

16 [C++] public: const int HeaderContentType;

17 [VB] Public Const HeaderContentType As Integer

18 [JScript] public var HeaderContentType : int;

19  
20 *Description*

21  
22 ToString

23  
24 [C#] public const int HeaderCookie;

25 [C++] public: const int HeaderCookie;

1	[VB]	Public	Const	HeaderCookie	As	Integer
2	[JScript]	public	var	HeaderCookie	:	int;

3

4 *Description*

5

6 ToString

7

8	[C#]	public	const	int	HeaderDate;
---	------	--------	-------	-----	-------------

9	[C++]	public:	const	int	HeaderDate;
---	-------	---------	-------	-----	-------------

10	[VB]	Public	Const	HeaderDate	As	Integer
----	------	--------	-------	------------	----	---------

11	[JScript]	public	var	HeaderDate	:	int;
----	-----------	--------	-----	------------	---	------

12

13 *Description*

14

15 ToString

16

17	[C#]	public	const	int	HeaderEtag;
----	------	--------	-------	-----	-------------

18	[C++]	public:	const	int	HeaderEtag;
----	-------	---------	-------	-----	-------------

19	[VB]	Public	Const	HeaderEtag	As	Integer
----	------	--------	-------	------------	----	---------

20	[JScript]	public	var	HeaderEtag	:	int;
----	-----------	--------	-----	------------	---	------

21

22 *Description*

23

24 ToString

25

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#]      public      const      int      HeaderExpect;
[C++]     public:     const      int      HeaderExpect;
[VB]      Public      Const      HeaderExpect      As      Integer
[JScript] public      var      HeaderExpect      :      int;
```

*Description*

ToString

```
[C#]      public      const      int      HeaderExpires;
[C++]     public:     const      int      HeaderExpires;
[VB]      Public      Const      HeaderExpires      As      Integer
[JScript] public      var      HeaderExpires      :      int;
```

*Description*

ToString

```
[C#]      public      const      int      HeaderFrom;
[C++]     public:     const      int      HeaderFrom;
[VB]      Public      Const      HeaderFrom      As      Integer
[JScript] public      var      HeaderFrom      :      int;
```

*Description*



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

ToString

[C#]	public	const	int	HeaderHost;
[C++]	public:	const	int	HeaderHost;
[VB]	Public	Const	HeaderHost	As Integer
[JScript]	public	var	HeaderHost	: int;

Description

ToString

[C#]	public	const	int	HeaderIfMatch;
[C++]	public:	const	int	HeaderIfMatch;
[VB]	Public	Const	HeaderIfMatch	As Integer
[JScript]	public	var	HeaderIfMatch	: int;

Description

ToString

[C#]	public	const	int	HeaderIfModifiedSince;
[C++]	public:	const	int	HeaderIfModifiedSince;
[VB]	Public	Const	HeaderIfModifiedSince	As Integer
[JScript]	public	var	HeaderIfModifiedSince	: int;

1  
2 *Description*

3  
4 ToString

5  
6 [C#] public const int HeaderIfNoneMatch;

7 [C++] public: const int HeaderIfNoneMatch;

8 [VB] Public Const HeaderIfNoneMatch As Integer

9 [JScript] public var HeaderIfNoneMatch : int;

10  
11 *Description*

12  
13 ToString

14  
15 [C#] public const int HeaderIfRange;

16 [C++] public: const int HeaderIfRange;

17 [VB] Public Const HeaderIfRange As Integer

18 [JScript] public var HeaderIfRange : int;

19  
20 *Description*

21  
22 ToString

23  
24 [C#] public const int HeaderIfUnmodifiedSince;

25 [C++] public: const int HeaderIfUnmodifiedSince;

1 [VB] Public Const HeaderIfUnmodifiedSince As Integer  
 2 [JScript] public var HeaderIfUnmodifiedSince : int;

4 *Description*

6 ToString

8 [C#] public const int HeaderKeepAlive;

9 [C++] public: const int HeaderKeepAlive;

10 [VB] Public Const HeaderKeepAlive As Integer

11 [JScript] public var HeaderKeepAlive : int;

13 *Description*

15 ToString

17 [C#] public const int HeaderLastModified;

18 [C++] public: const int HeaderLastModified;

19 [VB] Public Const HeaderLastModified As Integer

20 [JScript] public var HeaderLastModified : int;

22 *Description*

24 ToString

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#]      public      const      int      HeaderLocation;  
[C++]     public:     const      int      HeaderLocation;  
[VB]      Public      Const      HeaderLocation      As      Integer  
[JScript] public      var      HeaderLocation      :      int;
```

*Description*

ToString

```
[C#]      public      const      int      HeaderMaxForwards;  
[C++]     public:     const      int      HeaderMaxForwards;  
[VB]      Public      Const      HeaderMaxForwards      As      Integer  
[JScript] public      var      HeaderMaxForwards      :      int;
```

*Description*

ToString

```
[C#]      public      const      int      HeaderPragma;  
[C++]     public:     const      int      HeaderPragma;  
[VB]      Public      Const      HeaderPragma      As      Integer  
[JScript] public      var      HeaderPragma      :      int;
```

*Description*

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

ToString

```
[C#]      public      const      int      HeaderProxyAuthenticate;  
[C++]     public:     const      int      HeaderProxyAuthenticate;  
[VB]      Public      Const      HeaderProxyAuthenticate      As      Integer  
[JScript] public      var      HeaderProxyAuthenticate      :      int;
```

Description

ToString

```
[C#]      public      const      int      HeaderProxyAuthorization;  
[C++]     public:     const      int      HeaderProxyAuthorization;  
[VB]      Public      Const      HeaderProxyAuthorization      As      Integer  
[JScript] public      var      HeaderProxyAuthorization      :      int;
```

Description

ToString

```
[C#]      public      const      int      HeaderRange;  
[C++]     public:     const      int      HeaderRange;  
[VB]      Public      Const      HeaderRange      As      Integer  
[JScript] public      var      HeaderRange      :      int;
```

1  
2 *Description*

3  
4 ToString

5  
6 [C#] public const int HeaderReferer;

7 [C++] public: const int HeaderReferer;

8 [VB] Public Const HeaderReferer As Integer

9 [JScript] public var HeaderReferer : int;

10  
11 *Description*

12  
13 ToString

14  
15 [C#] public const int HeaderRetryAfter;

16 [C++] public: const int HeaderRetryAfter;

17 [VB] Public Const HeaderRetryAfter As Integer

18 [JScript] public var HeaderRetryAfter : int;

19  
20 *Description*

21  
22 ToString

23  
24 [C#] public const int HeaderServer;

25 [C++] public: const int HeaderServer;

1	[VB]	Public	Const	HeaderServer	As	Integer
2	[JScript]	public	var	HeaderServer	:	int;

3

4 *Description*

5

6 ToString

7

8	[C#]	public	const	int	HeaderSetCookie;
---	------	--------	-------	-----	------------------

9	[C++]	public:	const	int	HeaderSetCookie;
---	-------	---------	-------	-----	------------------

10	[VB]	Public	Const	HeaderSetCookie	As	Integer
----	------	--------	-------	-----------------	----	---------

11	[JScript]	public	var	HeaderSetCookie	:	int;
----	-----------	--------	-----	-----------------	---	------

12

13 *Description*

14

15 ToString

16

17	[C#]	public	const	int	HeaderTe;
----	------	--------	-------	-----	-----------

18	[C++]	public:	const	int	HeaderTe;
----	-------	---------	-------	-----	-----------

19	[VB]	Public	Const	HeaderTe	As	Integer
----	------	--------	-------	----------	----	---------

20	[JScript]	public	var	HeaderTe	:	int;
----	-----------	--------	-----	----------	---	------

21

22 *Description*

23

24 ToString

25

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

[C#]	public	const	int	HeaderTrailer;
[C++]	public:	const	int	HeaderTrailer;
[VB]	Public	Const	HeaderTrailer	As Integer
[JScript]	public	var	HeaderTrailer	: int;

*Description*

ToString

[C#]	public	const	int	HeaderTransferEncoding;
[C++]	public:	const	int	HeaderTransferEncoding;
[VB]	Public	Const	HeaderTransferEncoding	As Integer
[JScript]	public	var	HeaderTransferEncoding	: int;

*Description*

ToString

[C#]	public	const	int	HeaderUpgrade;
[C++]	public:	const	int	HeaderUpgrade;
[VB]	Public	Const	HeaderUpgrade	As Integer
[JScript]	public	var	HeaderUpgrade	: int;

*Description*



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

ToString

[C#]	public	const	int	HeaderUserAgent;
[C++]	public:	const	int	HeaderUserAgent;
[VB]	Public	Const	HeaderUserAgent	As Integer
[JScript]	public	var	HeaderUserAgent	: int;

Description

ToString

[C#]	public	const	int	HeaderVary;
[C++]	public:	const	int	HeaderVary;
[VB]	Public	Const	HeaderVary	As Integer
[JScript]	public	var	HeaderVary	: int;

Description

ToString

[C#]	public	const	int	HeaderVia;
[C++]	public:	const	int	HeaderVia;
[VB]	Public	Const	HeaderVia	As Integer
[JScript]	public	var	HeaderVia	: int;

1  
2 *Description*

3  
4 ToString

5  
6 [C#] public const int HeaderWarning;

7 [C++] public: const int HeaderWarning;

8 [VB] Public Const HeaderWarning As Integer

9 [JScript] public var HeaderWarning : int;

10  
11 *Description*

12  
13 ToString

14  
15 [C#] public const int HeaderWwwAuthenticate;

16 [C++] public: const int HeaderWwwAuthenticate;

17 [VB] Public Const HeaderWwwAuthenticate As Integer

18 [JScript] public var HeaderWwwAuthenticate : int;

19  
20 *Description*

21  
22 ToString

23  
24 [C#] public const int ReasonCachePolicy;

25 [C++] public: const int ReasonCachePolicy;

1	[VB]	Public	Const	ReasonCachePolicy	As	Integer
2	[JScript]	public	var	ReasonCachePolicy	:	int;
3						
4	<i>Description</i>					
5						
6		ToString				
7						
8	[C#]	public	const	int	ReasonCacheSecurity;	
9	[C++]	public:	const	int	ReasonCacheSecurity;	
10	[VB]	Public	Const	ReasonCacheSecurity	As	Integer
11	[JScript]	public	var	ReasonCacheSecurity	:	int;
12						
13	<i>Description</i>					
14						
15		ToString				
16						
17	[C#]	public	const	int	ReasonClientDisconnect;	
18	[C++]	public:	const	int	ReasonClientDisconnect;	
19	[VB]	Public	Const	ReasonClientDisconnect	As	Integer
20	[JScript]	public	var	ReasonClientDisconnect	:	int;
21						
22	<i>Description</i>					
23						
24		ToString				
25						

1  
2 [C#] public const int ReasonDefault;  
3 [C++] public: const int ReasonDefault;  
4 [VB] Public Const ReasonDefault As Integer  
5 [JScript] public var ReasonDefault : int;

6  
7 *Description*

8  
9 ToString

10  
11 [C#] public const int ReasonFileHandleCacheMiss;  
12 [C++] public: const int ReasonFileHandleCacheMiss;  
13 [VB] Public Const ReasonFileHandleCacheMiss As Integer  
14 [JScript] public var ReasonFileHandleCacheMiss : int;

15  
16 *Description*

17  
18 ToString

19  
20 [C#] public const int ReasonResponseCacheMiss;  
21 [C++] public: const int ReasonResponseCacheMiss;  
22 [VB] Public Const ReasonResponseCacheMiss As Integer  
23 [JScript] public var ReasonResponseCacheMiss : int;

24  
25 *Description*

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

ToString

```
[C#]      public      const      int      RequestHeaderMaximum;  
[C++]     public:     const      int      RequestHeaderMaximum;  
[VB]      Public      Const      RequestHeaderMaximum      As      Integer  
[JScript] public      var      RequestHeaderMaximum      :      int;
```

Description

ToString

```
[C#]      public      const      int      ResponseHeaderMaximum;  
[C++]     public:     const      int      ResponseHeaderMaximum;  
[VB]      Public      Const      ResponseHeaderMaximum      As      Integer  
[JScript] public      var      ResponseHeaderMaximum      :      int;
```

Description

HttpWorkerRequest

Example Syntax:

ToString

```
[C#]      protected      HttpWorkerRequest();  
[C++]     protected:      HttpWorkerRequest();
```

```

1  [VB]                Protected                Sub                New()
2  [JScript] protected function HttpWorkerRequest();
3      MachineConfigPath
4      ToString
5
6  [C#]    public    virtual    string    MachineConfigPath    {get;}
7  [C++]   public:   __property    virtual    String*    get_MachineConfigPath();
8  [VB] Overridable Public ReadOnly Property MachineConfigPath As String
9  [JScript] public    function    get    MachineConfigPath()    :    String;
10
11 Description
12
13      MachineInstallDirectory
14      ToString
15
16 [C#]    public    virtual    string    MachineInstallDirectory    {get;}
17 [C++]   public:   __property    virtual    String*    get_MachineInstallDirectory();
18 [VB] Overridable Public ReadOnly Property MachineInstallDirectory As String
19 [JScript] public    function    get    MachineInstallDirectory()    :    String;
20
21 Description
22
23      CloseConnection
24
25 [C#]    public    virtual    void    CloseConnection();

```

1	[C++]	public:	virtual	void	CloseConnection();
2	[VB]	Overridable	Public	Sub	CloseConnection()
3	[JScript]	public	function		CloseConnection();

4

5 *Description*

6

7 EndOfRequest

8

9	[C#]	public	abstract	void	EndOfRequest();
10	[C++]	public:	virtual	void	EndOfRequest() = 0;
11	[VB]	MustOverride	Public	Sub	EndOfRequest()
12	[JScript]	public	abstract	function	EndOfRequest();

13

14 *Description*

15

16 FlushResponse

17

18	[C#]	public	abstract	void	FlushResponse(bool finalFlush);
19	[C++]	public:	virtual	void	FlushResponse(bool finalFlush) = 0;
20	[VB]	MustOverride	Public	Sub	FlushResponse(ByVal finalFlush As Boolean)
21	[JScript]	public	abstract	function	FlushResponse(finalFlush : Boolean);

22

23 *Description*

24

25 GetAppPath

```

1
2 [C#]      public      virtual      string      GetAppPath();
3 [C++]     public:     virtual      String*      GetAppPath();
4 [VB]      Overridable Public      Function      GetAppPath() As String
5 [JScript] public      function      GetAppPath() :      String;

```

#### 7 *Description*

8 When overridden in a derived class, returns the virtual path to the currently  
9 executing server application.

10 *Return Value:* The virtual path of the current application.

#### 11 *GetAppPathTranslated*

```

12
13 [C#]      public      virtual      string      GetAppPathTranslated();
14 [C++]     public:     virtual      String*      GetAppPathTranslated();
15 [VB]      Overridable Public      Function      GetAppPathTranslated() As String
16 [JScript] public      function      GetAppPathTranslated() :      String;

```

#### 18 *Description*

19 When overridden in a derived class, returns the UNC-translated path to the  
20 currently executing server application.

21 *Return Value:* The UNC physical path of the current application.

#### 22 *GetAppPoolID*

```

23
24 [C#]      public      virtual      string      GetAppPoolID();
25 [C++]     public:     virtual      String*      GetAppPoolID();

```



```

1  [VB]  Overridable  Public  Function  GetAppPoolID()  As  String
2  [JScript]  public  function  GetAppPoolID()  :  String;

```

3

#### 4 *Description*

5       When overridden in a derived class, returns the application pool ID for the  
6 current URL.

7 *Return Value:* Always returns null.

8       GetBytesRead

9

```

10 [C#]  public  virtual  long  GetBytesRead();

```

```

11 [C++]  public:  virtual  __int64  GetBytesRead();

```

```

12 [VB]  Overridable  Public  Function  GetBytesRead()  As  Long

```

```

13 [JScript]  public  function  GetBytesRead()  :  long;

```

14

#### 15 *Description*

16

17       GetClientCertificate

18

```

19 [C#]  public  virtual  byte[]  GetClientCertificate();

```

```

20 [C++]  public:  virtual  unsigned  char  GetClientCertificate()  __gc[];

```

```

21 [VB]  Overridable  Public  Function  GetClientCertificate()  As  Byte()

```

```

22 [JScript]  public  function  GetClientCertificate()  :  Byte[];

```

23

#### 24 *Description*

25

1 Defines the base worker class used by ASP.NET managed code for request  
2 processing.

### 3 GetClientCertificateBinaryIssuer

4  
5 [C#] public virtual byte[] GetClientCertificateBinaryIssuer();  
6 [C++] public: virtual unsigned char GetClientCertificateBinaryIssuer() \_\_gc[];  
7 [VB] Overridable Public Function GetClientCertificateBinaryIssuer() As Byte()  
8 [JScript] public function GetClientCertificateBinaryIssuer() : Byte[];

### 9 Description

### 10 GetClientCertificateEncoding

11  
12  
13  
14 [C#] public virtual int GetClientCertificateEncoding();  
15 [C++] public: virtual int GetClientCertificateEncoding();  
16 [VB] Overridable Public Function GetClientCertificateEncoding() As Integer  
17 [JScript] public function GetClientCertificateEncoding() : int;

### 18 Description

### 19 GetClientCertificatePublicKey

20  
21  
22  
23 [C#] public virtual byte[] GetClientCertificatePublicKey();  
24 [C++] public: virtual unsigned char GetClientCertificatePublicKey() \_\_gc[];  
25 [VB] Overridable Public Function GetClientCertificatePublicKey() As Byte()

1 [JScript] public function GetClientCertificatePublicKey() : Byte[];

2

3 *Description*

4

5 GetClientCertificateValidFrom

6

7 [C#] public virtual DateTime GetClientCertificateValidFrom();

8 [C++] public: virtual DateTime GetClientCertificateValidFrom();

9 [VB] Overridable Public Function GetClientCertificateValidFrom() As DateTime

10 [JScript] public function GetClientCertificateValidFrom() : DateTime;

11

12 *Description*

13

14 GetClientCertificateValidUntil

15

16 [C#] public virtual DateTime GetClientCertificateValidUntil();

17 [C++] public: virtual DateTime GetClientCertificateValidUntil();

18 [VB] Overridable Public Function GetClientCertificateValidUntil() As DateTime

19 [JScript] public function GetClientCertificateValidUntil() : DateTime;

20

21 *Description*

22

23 GetConnectionID

24

25 [C#] public virtual long GetConnectionID();

```

1  [C++]      public:      virtual      __int64      GetConnectionID();
2  [VB]      Overridable  Public  Function  GetConnectionID()  As  Long
3  [JScript]  public      function      GetConnectionID()      :      long;

```

#### 5 *Description*

6 When overridden in a derived class, returns the ID of the current  
7 connection.

8 *Return Value:* Always returns 0.

#### 9 *GetFilePath*

```

10
11 [C#]      public      virtual      string      GetFilePath();
12 [C++]      public:      virtual      String*      GetFilePath();
13 [VB]      Overridable  Public  Function  GetFilePath()  As  String
14 [JScript]  public      function      GetFilePath()      :      String;

```

#### 16 *Description*

17 When overridden in a derived class, returns the physical path to the  
18 requested URI.

19 *Return Value:* The path to the URI.

#### 20 *GetFilePathTranslated*

```

21
22 [C#]      public      virtual      string      GetFilePathTranslated();
23 [C++]      public:      virtual      String*      GetFilePathTranslated();
24 [VB]      Overridable  Public  Function  GetFilePathTranslated()  As  String
25 [JScript]  public      function      GetFilePathTranslated()      :      String;

```

## Description

When overridden in a derived class, returns the translated file path to the requested URI (from virtual path to UNC path, ie "/proj1/page.aspx" to "c:\dir\page.aspx")

*Return Value:* The translated file path.

### GetHttpVerbName

```
[C#]      public      abstract      string      GetHttpVerbName();
[C++]     public:     virtual      String*      GetHttpVerbName()      =      0;
[VB]      MustOverride Public Function GetHttpVerbName() As String
[JScript] public      abstract      function      GetHttpVerbName()      :      String;
```

## Description

Provides access to the specified member of the request header.

*Return Value:* The HTTP verb returned in the request header.

### GetHttpVersion

```
[C#]      public      abstract      string      GetHttpVersion();
[C++]     public:     virtual      String*      GetHttpVersion()      =      0;
[VB]      MustOverride Public Function GetHttpVersion() As String
[JScript] public      abstract      function      GetHttpVersion()      :      String;
```

## Description

Provides access to the specified member of the request header.

*Return Value:* The HTTP version returned in the request header.

### GetKnownRequestHeader

[C#] public virtual string GetKnownRequestHeader(int index);

[C++] public: virtual String\* GetKnownRequestHeader(int index);

[VB] Overridable Public Function GetKnownRequestHeader(ByVal index As Integer) As String

[JScript] public function GetKnownRequestHeader(index : int) : String;

### *Description*

### GetKnownRequestHeaderIndex

[C#] - public static int GetKnownRequestHeaderIndex(string header);

[C++] public: static int GetKnownRequestHeaderIndex(String\* header);

[VB] Public Shared Function GetKnownRequestHeaderIndex(ByVal header As String) As Integer

[JScript] public static function GetKnownRequestHeaderIndex(header : String) : int;

### *Description*

### GetKnownRequestHeaderName

```

1
2 [C#] public static string GetKnownRequestHeaderName(int index);
3 [C++] public: static String* GetKnownRequestHeaderName(int index);
4 [VB] Public Shared Function GetKnownRequestHeaderName(ByVal index As
5 Integer) As String
6 [JScript] public static function GetKnownRequestHeaderName(index : int) :
7 String;
8

```

### *Description*

#### GetKnownResponseHeaderIndex

```

12
13 [C#] public static int GetKnownResponseHeaderIndex(string header);
14 [C++] public: static int GetKnownResponseHeaderIndex(String* header);
15 [VB] Public Shared Function GetKnownResponseHeaderIndex(ByVal header As
16 String) As Integer
17 [JScript] public static function GetKnownResponseHeaderIndex(header : String) :
18 int;
19

```

### *Description*

#### GetKnownResponseHeaderName

```

23
24 [C#] public static string GetKnownResponseHeaderName(int index);
25 [C++] public: static String* GetKnownResponseHeaderName(int index);

```

```

1 [VB] Public Shared Function GetKnownResponseHeaderName(ByVal index As
2 Integer) As String
3 [JScript] public static function GetKnownResponseHeaderName(index : int) :
4 String;

```

#### 6 *Description*

#### 8 GetLocalAddress

```

9
10 [C#] public abstract string GetLocalAddress();
11 [C++] public: virtual String* GetLocalAddress() = 0;
12 [VB] MustOverride Public Function GetLocalAddress() As String
13 [JScript] public abstract function GetLocalAddress() : String;

```

#### 15 *Description*

16 Provides access to the specified member of the request header.

17 *Return Value:* The server's IP address returned in the request header.

#### 18 GetLocalPort

```

19
20 [C#] public abstract int GetLocalPort();
21 [C++] public: virtual int GetLocalPort() = 0;
22 [VB] MustOverride Public Function GetLocalPort() As Integer
23 [JScript] public abstract function GetLocalPort() : int;

```

#### 25 *Description*



Provides access to the specified member of the request header.

*Return Value:* The server's port number returned in the request header.

### GetPathInfo

[C#]            public            virtual            string            GetPathInfo();

[C++]           public:           virtual           String\*           GetPathInfo();

[VB]    Overridable    Public    Function    GetPathInfo()    As    String

[JScript]       public       function       GetPathInfo()       :       String;

### Description

When overridden in a derived class, returns additional path information for a resource with a URL extension. That is, for the URL /virdir/page.html/tail, the

PathInfo                            value                            is                            /tail.

*Return Value:* Additional path information for a resource.

### GetPreloadedEntityBody

[C#]            public            virtual            byte[]            GetPreloadedEntityBody();

[C++]           public:           virtual           unsigned char    GetPreloadedEntityBody()    \_\_gc[];

[VB]    Overridable    Public    Function    GetPreloadedEntityBody()    As    Byte()

[JScript]       public       function       GetPreloadedEntityBody()       :       Byte[];

### Description

### GetProtocol

```

1
2 [C#]      public      virtual      string      GetProtocol();
3 [C++]     public:     virtual      String*      GetProtocol();
4 [VB]      Overridable Public      Function      GetProtocol() As String
5 [JScript] public      function      GetProtocol() :      String;
6

```

### 7 *Description*

8 When overridden in a derived class, returns the HTTP protocol (HTTP or  
9 HTTPS).

10 *Return Value:* HTTPS if IsSecure is **true** , otherwise **false** .

### 11 *GetQueryString*

```

12
13 [C#]      public      abstract      string      GetQueryString();
14 [C++]     public:     virtual      String*      GetQueryString() = 0;
15 [VB]      MustOverride Public      Function      GetQueryString() As String
16 [JScript] public      abstract      function      GetQueryString() : String;
17

```

### 18 *Description*

19 Provides access to the specified member of the request header.

20 *Return Value:* The QueryString member of the request header.

### 21 *GetQueryStringRawBytes*

```

22
23 [C#]      public      virtual      byte[]      GetQueryStringRawBytes();
24 [C++]     public:     virtual      unsigned char GetQueryStringRawBytes() __gc[];
25 [VB]      Overridable Public      Function      GetQueryStringRawBytes() As Byte()

```

1 [JScript] public function GetQueryStringRawBytes() : Byte[];

3 *Description*

4 When overridden in a derived class, returns the response query string as an  
5 array of bytes.

6 *Return Value:* An array of bytes containing the response.

7 GetRawUrl

9 [C#] public abstract string GetRawUrl();

10 [C++] public: virtual String\* GetRawUrl() = 0;

11 [VB] MustOverride Public Function GetRawUrl() As String

12 [JScript] public abstract function GetRawUrl() : String;

14 *Description*

15 Provides access to the specified member of the request header.

16 *Return Value:* The raw url member of the request header.

17 GetRemoteAddress

19 [C#] public abstract string GetRemoteAddress();

20 [C++] public: virtual String\* GetRemoteAddress() = 0;

21 [VB] MustOverride Public Function GetRemoteAddress() As String

22 [JScript] public abstract function GetRemoteAddress() : String;

24 *Description*

Provides access to the specified member of the request header.

*Return Value:* The client's IP address returned in the request header.

### GetRemoteName

[C#] public virtual string GetRemoteName();

[C++] public: virtual String\* GetRemoteName();

[VB] Overridable Public Function GetRemoteName() As String

[JScript] public function GetRemoteName() : String;

### Description

When overridden in a derived class, returns the client computer's name.

*Return Value:* The name of the client machine.

### GetRemotePort

[C#] public abstract int GetRemotePort();

[C++] public: virtual int GetRemotePort() = 0;

[VB] MustOverride Public Function GetRemotePort() As Integer

[JScript] public abstract function GetRemotePort() : int;

### Description

Provides access to the specified member of the request header.

*Return Value:* The client HTTP port number returned in the request header.

### GetRequestReason

[C#] public virtual int GetRequestReason();

```

1  [C++]      public:      virtual      int      GetRequestReason();
2  [VB]  Overridable  Public  Function  GetRequestReason()  As  Integer
3  [JScript]  public      function      GetRequestReason()      :      int;
4

```

#### 5 *Description*

6 When overridden in a derived class, returns the reason for the request.

7 *Return Value:* Reason code. The default is ReasonResponseCacheMiss.

#### 8 *GetServerName*

```

9
10 [C#]      public      virtual      string      GetServerName();
11 [C++]      public:      virtual      String*      GetServerName();
12 [VB]  Overridable  Public  Function  GetServerName()  As  String
13 [JScript]  public      function      GetServerName()      :      String;
14

```

#### 15 *Description*

16 When overridden in a derived class, returns the name of the local server..

17 *Return Value:* The name of the server.

#### 18 *GetServerVariable*

```

19
20 [C#]      public      virtual      string      GetServerVariable(string      name);
21 [C++]      public:      virtual      String*      GetServerVariable(String*      name);
22 [VB]  Overridable  Public  Function  GetServerVariable(ByVal name As String) As
23 String
24 [JScript]  public      function      GetServerVariable(name      :      String)      :      String;
25

```

1  
2 *Description*

3  
4       GetStatusDescription

5  
6 [C#]     public     static     string     GetStatusDescription(int     code);

7 [C++]    public:     static     String\*    GetStatusDescription(int     code);

8 [VB] Public Shared Function GetStatusDescription(ByVal code As Integer) As  
9 String

10 [JScript] public static function GetStatusDescription(code : int) : String;

11  
12 *Description*

13  
14       GetUnknownRequestHeader

15  
16 [C#]     public     virtual   string     GetUnknownRequestHeader(string   name);

17 [C++]    public:     virtual   String\*   GetUnknownRequestHeader(String\*   name);

18 [VB] Overridable Public Function GetUnknownRequestHeader(ByVal name As  
19 String)                               As                               String

20 [JScript] public function GetUnknownRequestHeader(name : String) : String;

21  
22 *Description*

23  
24       GetUnknownRequestHeaders

```

1
2 [C#]      public      virtual      string[][]      GetUnknownRequestHeaders();
3 [C++]      public:     virtual      String*      GetUnknownRequestHeaders()      [];
4 [VB] Overridable Public Function GetUnknownRequestHeaders() As String()
5 [JScript] public function GetUnknownRequestHeaders() : String[][];
6

```

### *Description*

#### GetUriPath

```

11 [C#]      public      abstract      string      GetUriPath();
12 [C++]      public:     virtual      String*      GetUriPath()      =      0;
13 [VB] MustOverride Public Function GetUriPath() As String
14 [JScript] public abstract function GetUriPath() : String;
15

```

### *Description*

Returns the physical path to the requested URI.

*Return Value:* The path of the requested URI.

#### GetUrlContextID

```

21 [C#]      public      virtual      long      GetUrlContextID();
22 [C++]      public:     virtual      __int64      GetUrlContextID();
23 [VB] Overridable Public Function GetUrlContextID() As Long
24 [JScript] public function GetUrlContextID() : long;
25

```

## Description

When overridden in a derived class, returns the context ID of the current connection.

*Return Value:* Always returns 0.

### GetUserToken

```
[C#]      public      virtual      IntPtr      GetUserToken();
[C++]      public:      virtual      IntPtr      GetUserToken();
[VB]      Overridable  Public  Function  GetUserToken()  As  IntPtr
[JScript]  public      function  GetUserToken()      :      IntPtr;
```

## Description

When overridden in a derived class, returns the client's impersonation token.

*Return Value:* The value representing the impersonation token. The default is 0.

### GetVirtualPathToken

```
[C#]      public      virtual      IntPtr      GetVirtualPathToken();
[C++]      public:      virtual      IntPtr      GetVirtualPathToken();
[VB]      Overridable  Public  Function  GetVirtualPathToken()  As  IntPtr
[JScript]  public      function  GetVirtualPathToken()      :      IntPtr;
```

## Description



1           HasEntityBody

2

3   [C#]           public           bool           HasEntityBody();

4   [C++]           public:           bool           HasEntityBody();

5   [VB]    Public    Function   HasEntityBody()   As    Boolean

6   [JScript]   public    function   HasEntityBody()   :    Boolean;

7

8   *Description*

9

10           HeadersSent

11

12   [C#]           public           virtual           bool           HeadersSent();

13   [C++]           public:           virtual           bool           HeadersSent();

14   [VB]    Overridable   Public    Function   HeadersSent()   As    Boolean

15   [JScript]   public    function   HeadersSent()   :    Boolean;

16

17   *Description*

18

19           IsClientConnected

20

21   [C#]           public           virtual           bool           IsClientConnected();

22   [C++]           public:           virtual           bool           IsClientConnected();

23   [VB]    Overridable   Public    Function   IsClientConnected()   As    Boolean

24   [JScript]   public    function   IsClientConnected()   :    Boolean;

25

1  
2 *Description*

3  
4 **IsEntireEntityBodyIsPreloaded**

5  
6 [C#] public virtual bool IsEntireEntityBodyIsPreloaded();

7 [C++] public: virtual bool IsEntireEntityBodyIsPreloaded();

8 [VB] Overridable Public Function IsEntireEntityBodyIsPreloaded() As Boolean

9 [JScript] public function IsEntireEntityBodyIsPreloaded() : Boolean;

10  
11 *Description*

12  
13 **IsSecure**

14  
15 [C#] public virtual bool IsSecure();

16 [C++] public: virtual bool IsSecure();

17 [VB] Overridable Public Function IsSecure() As Boolean

18 [JScript] public function IsSecure() : Boolean;

19  
20 *Description*

21 When overridden in a derived class, returns a value indicating whether the  
22 connection is secure (using SSL).

23 *Return Value:* **true** if the connection is secure, otherwise **false** . The default is  
24 **false** .

25 **MapPath**

```

1
2 [C#]      public      virtual      string      MapPath(string      virtualPath);
3 [C++]      public:      virtual      String*      MapPath(String*      virtualPath);
4 [VB] Overridable Public Function MapPath(ByVal virtualPath As String) As
5 String
6 [JScript] public function MapPath(virtualPath : String) : String;
7

```

### *Description*

#### ReadEntityBody

```

12 [C#]      public      virtual      int      ReadEntityBody(byte[]      buffer,      int      size);
13 [C++]      public:      virtual      int      ReadEntityBody(unsigned char buffer __gc[], int size);
14 [VB] Overridable Public Function ReadEntityBody(ByVal buffer() As Byte,
15 ByVal      size      As      Integer)      As      Integer
16 [JScript] public function ReadEntityBody(buffer : Byte[], size : int) : int;
17

```

### *Description*

#### SendCalculatedContentLength

```

22 [C#]      public      virtual      void      SendCalculatedContentLength(int      contentLength);
23 [C++]      public:      virtual      void      SendCalculatedContentLength(int      contentLength);
24 [VB] Overridable Public Sub SendCalculatedContentLength(ByVal contentLength
25 As      Integer)

```

1 [JScript] public function SendCalculatedContentLength(contentLength : int);

2  
3 *Description*

4  
5 SendKnownResponseHeader

6  
7 [C#] public abstract void SendKnownResponseHeader(int index, string value);

8 [C++] public: virtual void SendKnownResponseHeader(int index, String\* value) =  
9 0;

10 [VB] MustOverride Public Sub SendKnownResponseHeader(ByVal index As  
11 Integer, ByVal value As String)

12 [JScript] public abstract function SendKnownResponseHeader(index : int, value :  
13 String);

14  
15 *Description*

16  
17 SendResponseFromFile

18  
19 [C#] public abstract void SendResponseFromFile(IntPtr handle, long offset, long  
20 length);

21 [C++] public: virtual void SendResponseFromFile(IntPtr handle, \_\_int64 offset,  
22 \_\_int64 length) = 0;

23 [VB] MustOverride Public Sub SendResponseFromFile(ByVal handle As IntPtr,  
24 ByVal offset As Long, ByVal length As Long)

25 [JScript] public abstract function SendResponseFromFile(handle : IntPtr, offset :

1 long, length : long);

2

3 *Description*

4

5 SendResponseFromFile

6

7 [C#] public abstract void SendResponseFromFile(string filename, long offset, long  
8 length);

9 [C++] public: virtual void SendResponseFromFile(String\* filename, \_\_int64  
10 offset, \_\_int64 length) = 0;

11 [VB] MustOverride Public Sub SendResponseFromFile(ByVal filename As  
12 String, ByVal offset As Long, ByVal length As Long)

13 [JScript] public abstract function SendResponseFromFile(filename : String, offset  
14 : long, length : long);

15

16 *Description*

17

18 SendResponseFromMemory

19

20 [C#] public abstract void SendResponseFromMemory(byte[] data, int length);

21 [C++] public: virtual void SendResponseFromMemory(unsigned char data \_\_gc[],  
22 int length) = 0;

23 [VB] MustOverride Public Sub SendResponseFromMemory(ByVal data() As  
24 Byte, ByVal length As Integer)

25 [JScript] public abstract function SendResponseFromMemory(data : Byte[], length

1 : int);

2  
3 *Description*

4  
5 **SendResponseFromMemory**

6  
7 [C#] public virtual void SendResponseFromMemory(IntPtr data, int length);

8 [C++] public: virtual void SendResponseFromMemory(IntPtr data, int length);

9 [VB] Overridable Public Sub SendResponseFromMemory(ByVal data As IntPtr,

10 ByVal length As Integer)

11 [JScript] public function SendResponseFromMemory(data : IntPtr, length : int);

12  
13 *Description*

14  
15 **SendStatus**

16  
17 [C#] public abstract void SendStatus(int statusCode, string statusDescription);

18 [C++] public: virtual void SendStatus(int statusCode, String\* statusDescription) =

19 0;

20 [VB] MustOverride Public Sub SendStatus(ByVal statusCode As Integer, ByVal

21 statusDescription As String)

22 [JScript] public abstract function SendStatus(statusCode : int, statusDescription :

23 String);

24  
25 *Description*

## SendUnknownResponseHeader

[C#] public abstract void SendUnknownResponseHeader(string name, string value);

[C++] public: virtual void SendUnknownResponseHeader(String\* name, String\* value) = 0;

[VB] MustOverride Public Sub SendUnknownResponseHeader(ByVal name As String, ByVal value As String)

[JScript] public abstract function SendUnknownResponseHeader(name : String, value : String);

## Description

## SetEndOfSendNotification

[C#] public virtual void SetEndOfSendNotification(HttpWorkerRequest.EndOfSendNotification callback, object extraData);

[C++] public: virtual void SetEndOfSendNotification(HttpWorkerRequest.EndOfSendNotification\* callback, Object\* extraData);

[VB] Overridable Public Sub SetEndOfSendNotification(ByVal callback As HttpWorkerRequest.EndOfSendNotification, ByVal extraData As Object)

[JScript] public function SetEndOfSendNotification(callback :

1 `HttpWorkerRequest.EndOfSendNotification,      extraData      :      Object);`

2  
3 *Description*

4  
5 `HttpWriter` class (`System.Web`)

6 `ToString`

7  
8  
9 *Description*

10 Provides a **`System.IO.TextWriter`** object that is accessed through the  
11 intrinsic **`System.Web.HttpResponse`** object.

12 The **`System.Web.HttpResponse.Write(System.String)`** methods of the  
13 intrinsic **`System.Web.HttpResponse`** object make internal calls to an **`HttpWriter`**  
14 object.

15 `Encoding`

16 `ToString`

17  
18  
19 *Description*

20 Gets an **`System.Text.Encoding`** object for the **`System.IO.TextWriter`** .

21 `FormatProvider`

22 `NewLine`

23 `OutputStream`

24 `ToString`



1  
2  
3 *Description*

4 Gets a **System.IO.Stream** object to enable HTTP output directly from the  
5 **Stream** .

6 Close

7  
8 [C#] public override void Close();  
9 [C++] public: void Close();  
10 [VB] Overrides Public Sub Close()  
11 [JScript] public override function Close();  
12

13 *Description*

14 Sends all buffered output to the HTTP output stream and closes the socket  
15 connection.

16 Flush

17  
18 [C#] public override void Flush();  
19 [C++] public: void Flush();  
20 [VB] Overrides Public Sub Flush()  
21 [JScript] public override function Flush();  
22

23 *Description*

24 Sends all buffered output to the HTTP output stream.

25 Write

1  
2 [C#] public override void Write(char ch);  
3 [C++] public: void Write(\_\_wchar\_t ch);  
4 [VB] Overrides Public Sub Write(ByVal ch As Char)  
5 [JScript] public override function Write(ch : Char); Sends HTTP output to the  
6 client.

7  
8 *Description*

9 Sends a single character to the HTTP output stream. The character to send  
10 to the HTTP output stream.

11 Write

12  
13 [C#] public override void Write(object obj);  
14 [C++] public: void Write(Object\* obj);  
15 [VB] Overrides Public Sub Write(ByVal obj As Object)  
16 [JScript] public override function Write(obj : Object);

17  
18 *Description*

19 Sends an **Object** to the HTTP output stream. The **Object** to send to the  
20 HTTP output stream.

21 Write

22  
23 [C#] public override void Write(string s);  
24 [C++] public: void Write(String\* s);  
25 [VB] Overrides Public Sub Write(ByVal s As String)

1 [JScript] public override function Write(s : String);

3 *Description*

4 Sends a string to the HTTP output stream. The string to send to the HTTP  
5 output stream.

6 Write

8 [C#] public override void Write(char[] buffer, int index, int count);

9 [C++] public: void Write(\_\_wchar\_t buffer \_\_gc[], int index, int count);

10 [VB] Overrides Public Sub Write(ByVal buffer() As Char, ByVal index As  
11 Integer, ByVal count As Integer)

12 [JScript] public override function Write(buffer : Char[], index : int, count : int);

14 *Description*

15 Sends a stream of characters with the specified starting position and  
16 number of characters to the HTTP output stream. The memory buffer containing  
17 the characters to send to the HTTP output stream The buffer position of the first  
18 character to send. The number of characters to send beginning at the position  
19 specified by *index*.

20 WriteBytes

22 [C#] public void WriteBytes(byte[] buffer, int index, int count);

23 [C++] public: void WriteBytes(unsigned char buffer \_\_gc[], int index, int count);

24 [VB] Public Sub WriteBytes(ByVal buffer() As Byte, ByVal index As Integer,  
25 ByVal count As Integer)

1 [JScript] public function WriteBytes(buffer : Byte[], index : int, count : int);

2  
3 *Description*

4 Sends a stream of bytes with the specified starting position and number of  
5 bytes to the HTTP output stream. The memory buffer containing the bytes to send  
6 to the HTTP output stream. The buffer position of the first byte to send. The  
7 number of bytes to send, beginning at the byte position specified by *index* .

8 WriteLine

9  
10 [C#] public override void WriteLine();

11 [C++] public: void WriteLine();

12 [VB] Overrides Public Sub WriteLine()

13 [JScript] public override function WriteLine();

14  
15 *Description*

16 Sends a carriage return + line feed (CRLF) pair of characters to the HTTP  
17 output stream.

18 WriteString

19  
20 [C#] public void WriteString(string s, int index, int count);

21 [C++] public: void WriteString(String\* s, int index, int count);

22 [VB] Public Sub WriteString(ByVal s As String, ByVal index As Integer, ByVal  
23 count As Integer)

24 [JScript] public function WriteString(s : String, index : int, count : int);

1  
2 *Description*

3 Sends a string or a sub-string to the client.

4 IHttpAsyncHandler interface (System.Web)

5 WriteString

6  
7  
8 *Description*

9 When implemented by a class, defines the contract that HTTP  
10 asynchronous handler objects must implement.

11 BeginProcessRequest

12  
13 [C#] IAsyncResult BeginProcessRequest(HttpContext context, AsyncCallback cb,  
14 object extraData);

15 [C++] IAsyncResult\* BeginProcessRequest(HttpContext\* context,  
16 AsyncCallback\* cb, Object\* extraData);

17 [VB] Function BeginProcessRequest(ByVal context As HttpContext, ByVal cb As  
18 AsyncCallback, ByVal extraData As Object) As IAsyncResult

19 [JScript] function BeginProcessRequest(context : HttpContext, cb :  
20 AsyncCallback, extraData : Object) : IAsyncResult;

21  
22 *Description*

23 Initiates an asynchronous call to the HTTP handler.

24 *Return Value:* An **System.IAsyncResult** that contains information about the status  
25 of the process. An **System.Web.HttpContext** object that provides references to

intrinsic server objects (for example, **Request**, **Response**, **Session**, and **Server**) used to service HTTP requests. The **System.AsyncCallback** to call when the asynchronous method call is complete. If *cb* is a null reference (**Nothing** in Visual Basic), the delegate is not called. Any extra data needed to process the request.

**EndProcessRequest**

[C#]	void	EndProcessRequest(IAsyncResult	result);
[C++]	void	EndProcessRequest(IAsyncResult*	result);
[VB]	Sub	EndProcessRequest(ByVal result As	IAsyncResult)
[JScript]	function	EndProcessRequest(result	: IAsyncResult);

### *Description*

Executes clean-up code when the process ends. An **System.IAsyncResult** that contains information about the status of the process.

**IHttpHandler** interface (System.Web)

**EndProcessRequest**

### *Description*

Defines the contract that ASP.NET implements to synchronously process HTTP Web requests using custom HTTP handlers.

You can write custom HTTP handlers to process specific, predefined types of HTTP requests in any Common Language Specification (CLS) compliant language. Executable code defined in the **HttpHandler** classes, rather than conventional ASP or ASP.NET Web pages, responds to these specific requests.

HTTP handlers give you a means of interacting with the low-level request and response services of the IIS Web server and provide functionality much like ISAPI extensions but with a simpler programming model.

**IsReusable**

**EndProcessRequest**

[C#]	bool	IsReusable	{get;}
[C++]	bool	get_IsReusable();	
[VB]	ReadOnly Property	IsReusable	As Boolean
[JScript]	abstract function	get IsReusable()	: Boolean;

#### *Description*

Gets a value indicating whether another request can use the **System.Web.IHttpHandler** instance.

You explicitly set the **IsReusable** property to **true** or **false** by code you provide that overrides the **IsReusable** property accessor.

**ProcessRequest**

[C#]	void	ProcessRequest(HttpContext	context);
[C++]	void	ProcessRequest(HttpContext*	context);
[VB]	Sub	ProcessRequest(ByVal context	As HttpContext)
[JScript]	function	ProcessRequest(context	: HttpContext);

#### *Description*

1 Enables processing of HTTP Web requests by a custom **HttpHandler** that  
2 implements the **System.Web.IHttpHandler** interface.

3 Place your custom **HttpHandler** code in the **ProcessRequest** virtual  
4 method as shown in the following example. An **System.Web.HttpContext** object  
5 that provides references to the intrinsic server objects (for example, **Request**,  
6 **Response**, **Session**, and **Server**) used to service HTTP requests.

7 IHttpHandlerFactory interface (System.Web)

8 ProcessRequest

9  
10  
11 *Description*

12 Defines the contract that class factories must implement to create new  
13 **System.Web.IHttpHandler** objects.

14 A class that implements the **IHttpHandlerFactory** interface has no  
15 behavior except to dynamically manufacture new handler objects--that is, new  
16 instances of classes that implement the **IHttpHandler** interface.

17 GetHandler

18  
19 [C#] IHttpHandler GetHandler(HttpContext context, string requestType, string url,  
20 string pathTranslated);

21 [C++] IHttpHandler\* GetHandler(HttpContext\* context, String\* requestType,  
22 String\* url, String\* pathTranslated);

23 [VB] Function GetHandler(ByVal context As HttpContext, ByVal requestType As  
24 String, ByVal url As String, ByVal pathTranslated As String) As IHttpHandler

25 [JScript] function GetHandler(context : HttpContext, requestType : String, url :



String, pathTranslated : String) : IHttpHandler;

### Description

Returns an instance of a class that implements the IHttpHandler interface.

**Return Value:** A new **IHttpHandler** object that processes the request. An instance of the **System.Web.HttpContext** class that provides references to intrinsic server objects (For example, **Request**, **Response**, **Session**, and **Server**) used to service HTTP requests. The HTTP data transfer method (**GET** or **POST**) that the client uses. The **System.Web.HttpRequest.RawUrl** of the requested resource. The **System.Web.HttpRequest.PhysicalApplicationPath** to the requested resource.

### ReleaseHandler

[C#] void ReleaseHandler(IHttpHandler handler);

[C++] void ReleaseHandler(IHttpHandler\* handler);

[VB] Sub ReleaseHandler(ByVal handler As IHttpHandler)

[JScript] function ReleaseHandler(handler : IHttpHandler);

### Description

Enables a factory to reuse an existing handler instance. The IHttpHandler object to reuse.

IHttpModule interface (System.Web)

### ReleaseHandler

### Description

Provides module initialization and disposal events to the inheriting class.

## Dispose

[C#]	void	Dispose();
[C++]	void	Dispose();
[VB]	Sub	Dispose()
[JScript]	function	Dispose();

### *Description*

Disposes of the resources (other than memory) used by the module that implements **IHttpModule**.

**Dispose** performs any final cleanup work prior to removal of the module from the execution pipeline.

## Init

[C#]	void	Init(HttpApplication	context);
[C++]	void	Init(HttpApplication*	context);
[VB]	Sub	Init(ByVal context	As HttpApplication)
[JScript]	function	Init(context	: HttpApplication);

### *Description*

Initializes a module and prepares it to handle requests.

ProcessInfo class (System.Web)

## Init

1  
2  
3 *Description*

4 Provides information on processes currently executing.

5 ProcessInfo

6 *Example Syntax:*

7 Init

8  
9 [C#] public ProcessInfo();  
10 [C++] public: ProcessInfo();  
11 [VB] Public Sub New()  
12 [JScript] public function ProcessInfo();  
13

14 *Description*

15 Initializes a new instance of the **System.Web.ProcessInfo** class.

16 The default constructor initializes all fields to their default values.

17 ProcessInfo

18 *Example Syntax:*

19 Init

20  
21 [C#] public ProcessInfo(DateTime startTime, TimeSpan age, int processID, int  
22 requestCount, ProcessStatus status, ProcessShutdownReason shutdownReason, int  
23 peakMemoryUsed);

24 [C++] public: ProcessInfo(DateTime startTime, TimeSpan age, int processID, int  
25 requestCount, ProcessStatus status, ProcessShutdownReason shutdownReason, int

```

1 peakMemoryUsed);
2 [VB] Public Sub New(ByVal startTime As DateTime, ByVal age As TimeSpan,
3   ByVal processID As Integer, ByVal requestCount As Integer, ByVal status As
4   ProcessStatus, ByVal shutdownReason As ProcessShutdownReason, ByVal
5   peakMemoryUsed           As           Integer)
6 [JScript] public function ProcessInfo(startTime : DateTime, age : TimeSpan,
7   processID : int, requestCount : int, status : ProcessStatus, shutdownReason :
8   ProcessShutdownReason, peakMemoryUsed : int); Initializes a new instance of
9   the           System.Web.ProcessInfo           class.

```

#### 11 *Description*

12        Initializes a new instance of the **ProcessInfo** class and sets internal  
13 information indicating the status of the process. A **System.DateTime** that  
14 indicates the time at which the process started. The **System.TimeSpan** that  
15 indicates the time elapsed since the process started. The ID number assigned to the  
16 process. The number of start requests for the process. One of the  
17 **System.Web.ProcessStatus** values that indicates the current status of the process.  
18 One of the **System.Web.ProcessShutdownReason** values. The maximum  
19 memory used (in bytes).

20        Age

21        Init

```

22
23 [C#]           public           TimeSpan           Age           {get;}
24 [C++]          public:          __property         TimeSpan     get_Age();
25 [VB]           Public           ReadOnly           Property     Age           As           TimeSpan

```

1 [JScript] public function get Age() : TimeSpan;  
2

3 *Description*

4 Gets the length of time the process has been running.

5 PeakMemoryUsed

6 Init  
7

8 [C#] public int PeakMemoryUsed {get;}

9 [C++] public: \_\_property int get\_PeakMemoryUsed();

10 [VB] Public ReadOnly Property PeakMemoryUsed As Integer

11 [JScript] public function get PeakMemoryUsed() : int;  
12

13 *Description*

14 Gets the maximum amount of memory the process has used.

15 ProcessID

16 Init  
17

18 [C#] public int ProcessID {get;}

19 [C++] public: \_\_property int get\_ProcessID();

20 [VB] Public ReadOnly Property ProcessID As Integer

21 [JScript] public function get ProcessID() : int;  
22

23 *Description*

24 Gets the ID number assigned to the process.

25 RequestCount

## Init

```
[C#]          public          int          RequestCount          {get;}
[C++]         public:         __property    int          get_RequestCount();
[VB]   Public   ReadOnly   Property RequestCount   As   Integer
[JScript]     public   function   get   RequestCount()   :   int;
```

## Description

Gets the number of start requests for the process.

## ShutdownReason

### Init

```
[C#]          public          ProcessShutdownReason   ShutdownReason   {get;}
[C++]         public:         __property    ProcessShutdownReason   get_ShutdownReason();
[VB]   Public   ReadOnly   Property ShutdownReason   As   ProcessShutdownReason
[JScript]     public   function   get   ShutdownReason()   :   ProcessShutdownReason;
```

## Description

Gets a value that indicates why the process shut down.

## StartTime

### Init

```
[C#]          public          DateTime          StartTime          {get;}
[C++]         public:         __property    DateTime          get_StartTime();
[VB]   Public   ReadOnly   Property StartTime   As   DateTime
```

1 [JScript] public function get StartTime() : DateTime;

2

3 *Description*

4 Gets the time at which the process started.

5 Status

6 Init

7

8 [C#] public ProcessStatus Status {get;}

9 [C++] public: \_\_property ProcessStatus get\_Status();

10 [VB] Public ReadOnly Property Status As ProcessStatus

11 [JScript] public function get Status() : ProcessStatus;

12

13 *Description*

14 Gets the current status of the process.

15 SetAll

16

17 [C#] public void SetAll(DateTime startTime, TimeSpan age, int processID, int  
18 requestCount, ProcessStatus status, ProcessShutdownReason shutdownReason, int  
19 peakMemoryUsed);

20 [C++] public: void SetAll(DateTime startTime, TimeSpan age, int processID, int  
21 requestCount, ProcessStatus status, ProcessShutdownReason shutdownReason, int  
22 peakMemoryUsed);

23 [VB] Public Sub SetAll(ByVal startTime As DateTime, ByVal age As TimeSpan,  
24 ByVal processID As Integer, ByVal requestCount As Integer, ByVal status As  
25 ProcessStatus, ByVal shutdownReason As ProcessShutdownReason, ByVal

```

1 peakMemoryUsed As Integer)
2 [JScript] public function SetAll(startTime : DateTime, age : TimeSpan, processID
3 : int, requestCount : int, status : ProcessStatus, shutdownReason :
4 ProcessShutdownReason, peakMemoryUsed : int);
5

```

### *Description*

Sets internal information indicating the status of the process. A **System.DateTime** that indicates the time at which the process started. A **System.TimeSpan** that indicates the time elapsed since the process started. The ID number assigned to the process. The number of start requests for the process. One of the **System.Web.ProcessStatus** values that indicates the time elapsed since the process started. One of the **System.Web.ProcessShutdownReason** values. The maximum memory used (in bytes).

ProcessModelInfo class (System.Web)

ToString

ProcessModelInfo

### *Example Syntax:*

ToString

GetCurrentProcessInfo

```

21 [C#] public static ProcessInfo GetCurrentProcessInfo();

```

```

22 [C++] public: static ProcessInfo* GetCurrentProcessInfo();

```

```

23 [VB] Public Shared Function GetCurrentProcessInfo() As ProcessInfo

```

```

24 [JScript] public static function GetCurrentProcessInfo() : ProcessInfo;

```

GetHistory



```

1
2 [C#]    public    static    ProcessInfo[]    GetHistory(int    numRecords);
3 [C++]   public:   static    ProcessInfo*    GetHistory(int    numRecords)    [];
4 [VB]    Public    Shared    Function    GetHistory(ByVal    numRecords    As    Integer)    As
5 ProcessInfo()
6 [JScript] public static function GetHistory(numRecords : int) : ProcessInfo[];
7         ProcessShutdownReason enumeration (System.Web)
8         ToString
9
10

```

### *Description*

Provides enumerated values that indicate why a process has shut down.

ToString

```

15 [C#]    public    const    ProcessShutdownReason    IdleTimeout;
16 [C++]   public:   const    ProcessShutdownReason    IdleTimeout;
17 [VB]    Public    Const    IdleTimeout    As    ProcessShutdownReason
18 [JScript] public    var    IdleTimeout    :    ProcessShutdownReason;
19

```

### *Description*

Indicates that the process exceeded the allowable idle time.

ToString

```

24 [C#]    public    const    ProcessShutdownReason    MemoryLimitExceeded;
25 [C++]   public:   const    ProcessShutdownReason    MemoryLimitExceeded;

```

1 [VB] Public Const MemoryLimitExceeded As ProcessShutdownReason

2 [JScript] public var MemoryLimitExceeded : ProcessShutdownReason;

3  
4 *Description*

5 Indicates that the process exceeded the per-process memory limit.

6 ToString

7  
8 [C#] public const ProcessShutdownReason None;

9 [C++] public: const ProcessShutdownReason None;

10 [VB] Public Const None As ProcessShutdownReason

11 [JScript] public var None : ProcessShutdownReason;

12  
13 *Description*

14 Indicates that the process has not shut down.

15 ToString

16  
17 [C#] public const ProcessShutdownReason PingFailed;

18 [C++] public: const ProcessShutdownReason PingFailed;

19 [VB] Public Const PingFailed As ProcessShutdownReason

20 [JScript] public var PingFailed : ProcessShutdownReason;

21  
22 *Description*

23  
24 ToString

```

1
2 [C#]    public    const    ProcessShutdownReason    RequestQueueLimit;
3 [C++]   public:   const    ProcessShutdownReason    RequestQueueLimit;
4 [VB]    Public    Const    RequestQueueLimit    As    ProcessShutdownReason
5 [JScript] public  var    RequestQueueLimit    :    ProcessShutdownReason;
6

```

### *Description*

Indicates that the request for a process exceeded the allowable number of processes in the queue.

### *ToString*

```

12 [C#]    public    const    ProcessShutdownReason    RequestsLimit;
13 [C++]   public:   const    ProcessShutdownReason    RequestsLimit;
14 [VB]    Public    Const    RequestsLimit    As    ProcessShutdownReason
15 [JScript] public  var    RequestsLimit    :    ProcessShutdownReason;
16

```

### *Description*

Indicates that the request for the process exceeded the allowable number of processes.

### *ToString*

```

22 [C#]    public    const    ProcessShutdownReason    Timeout;
23 [C++]   public:   const    ProcessShutdownReason    Timeout;
24 [VB]    Public    Const    Timeout    As    ProcessShutdownReason
25 [JScript] public  var    Timeout    :    ProcessShutdownReason;

```

1  
2 *Description*

3 Indicates that the process timed out.

4 ToString

5  
6 [C#] public const ProcessShutdownReason Unexpected;

7 [C++] public: const ProcessShutdownReason Unexpected;

8 [VB] Public Const Unexpected As ProcessShutdownReason

9 [JScript] public var Unexpected : ProcessShutdownReason;

10  
11 *Description*

12 Indicates that the process shut down unexpectedly.

13 ProcessStatus enumeration (System.Web)

14 ToString

15  
16  
17 *Description*

18 Provides enumerated values that indicate the current status of a process.

19 ToString

20  
21 [C#] public const ProcessStatus Alive;

22 [C++] public: const ProcessStatus Alive;

23 [VB] Public Const Alive As ProcessStatus

24 [JScript] public var Alive : ProcessStatus;

*Description*

Indicates that the process is running.

ToString

[C#]            public            const            ProcessStatus            ShutDown;

[C++]            public:            const            ProcessStatus            ShutDown;

[VB]            Public            Const            ShutDown            As            ProcessStatus

[JScript]            public            var            ShutDown            :            ProcessStatus;

*Description*

Indicates that the process has shut down.

ToString

[C#]            public            const            ProcessStatus            ShuttingDown;

[C++]            public:            const            ProcessStatus            ShuttingDown;

[VB]            Public            Const            ShuttingDown            As            ProcessStatus

[JScript]            public            var            ShuttingDown            :            ProcessStatus;

*Description*

Indicates that the process has begun to shut down.

ToString

[C#]            public            const            ProcessStatus            Terminated;

[C++]            public:            const            ProcessStatus            Terminated;

1	[VB]	Public	Const	Terminated	As	ProcessStatus
2	[JScript]	public	var	Terminated	:	ProcessStatus;

3

4 *Description*

5 Indicates that the process has terminated.

6 TraceContext class (System.Web)

7 ToString

8

9

10 *Description*

11 Captures and presents execution details about a Web request. This class  
12 cannot be inherited.

13 You can use this class to append messages to specific trace categories. For  
14 example, if you are creating an instance of the  
15 **System.Web.UI.WebControls.Calendar** class for your application, you might  
16 include the trace message "Starting To Render" in a rendering procedure, and  
17 "Firing OnChange Event" in an event handling procedure.

18 TraceContext

19 *Example Syntax:*

20 ToString

21

22 [C#] public TraceContext(HttpContext context);

23 [C++] public: TraceContext(HttpContext\* context);

24 [VB] Public Sub New(ByVal context As HttpContext)

25 [JScript] public function TraceContext(context : HttpContext);

## *Description*

Initializes a new instance of the **System.Web.TraceContext** class. An **System.Web.HttpContext** that contains information about the current Web request.

IsEnabled

ToString

```
[C#]      public      bool      isEnabled      {get;      set;}
```

```
[C++] public: __property bool get_IsEnabled();public: __property void  
set_IsEnabled(bool);
```

```
[VB]      Public      Property      isEnabled      As      Boolean
```

```
[JScript] public function get isEnabled() : Boolean;public function set  
isEnabled(Boolean);
```

## *Description*

Indicates whether tracing is enabled for the current Web request.

Use this flag to check whether your page or application should output tracing information before it writes anything to the trace log. You can set this property to **true** for a page by including a trace="true" attribute in the directive. To set the property to **true** for an entire application, set it in the application's Web.config file.

TraceMode

ToString

```

1
2 [C#]      public      TraceMode      TraceMode      {get;      set;}
3 [C++] public: __property TraceMode get_TraceMode();public: __property void
4 set_TraceMode(TraceMode);
5 [VB]      Public      Property      TraceMode      As      TraceMode
6 [JScript] public function get TraceMode() : TraceMode;public function set
7 TraceMode(TraceMode);
8

```

### 9 *Description*

10 Gets or sets the sorted order in which trace messages should be output to a  
11 requesting browser.

12 Trace messages can be sorted in the order they were processed, or  
13 alphabetically by user-defined category.

### 14 **Warn**

```

15
16 [C#]      public      void      Warn(string      message);
17 [C++]      public:      void      Warn(String*      message);
18 [VB]      Public      Sub      Warn(ByVal      message      As      String)
19 [JScript] public function Warn(message : String); Writes trace information to the
20 trace log. Unlike System.Web.TraceContext.Write(System.String) , all
21 warnings appear in the log as red text.
22

```

### 23 *Description*

24 Writes a trace message to the trace log. All warnings appear as in the log as  
25 red text. The trace message to write to the log.



## Warn

```
[C#]    public    void    Warn(string    category,    string    message);  
[C++]   public:   void    Warn(String*    category,    String*    message);  
[VB]    Public Sub Warn(ByVal category As String, ByVal message As String)  
[JScript] public function Warn(category : String, message : String); Writes trace  
information to the trace log. All warnings appear in the log as red text.
```

### *Description*

Writes trace information to the trace log, including any user-defined categories and trace messages. All warnings appear in the log as red text. The trace category that receives the message. The trace message to write to the log.

## Warn

```
[C#]    public void Warn(string category, string message, Exception errorInfo);  
[C++]   public: void Warn(String* category, String* message, Exception*  
errorInfo);  
[VB]    Public Sub Warn(ByVal category As String, ByVal message As String,  
ByVal    errorInfo    As    Exception)  
[JScript] public function Warn(category : String, message : String, errorInfo :  
Exception);
```

### *Description*

Writes trace information to the trace log, including any user-defined categories, trace messages, and error information. All warnings appear in the log as

red text. The trace category that receives the message. The trace message to write to the log. An **System.Exception** that contains information about the error.

### Write

```
[C#]      public      void      Write(string      message);
[C++]     public:     void      Write(String*      message);
[VB]      Public      Sub      Write(ByVal      message      As      String)
[JScript] public function Write(message : String); Writes trace information to the
trace                                           log.
```

### *Description*

Writes a trace message to the trace log. The trace message to write to the log.

### Write

```
[C#]      public      void      Write(string      category,      string      message);
[C++]     public:     void      Write(String*      category,      String*      message);
[VB]      Public      Sub      Write(ByVal      category      As      String,      ByVal      message      As      String)
[JScript] public function Write(category : String, message : String); Writes trace
information      to      the      trace      log.
```

### *Description*

Writes trace information to the trace log, including a message and any user-defined categories. The trace category that receives the message. The trace message to write to the log.

Write

[C#] public void Write(string category, string message, Exception errorInfo);

[C++] public: void Write(String\* category, String\* message, Exception\* errorInfo);

[VB] Public Sub Write(ByVal category As String, ByVal message As String, ByVal errorInfo As Exception)

[JScript] public function Write(category : String, message : String, errorInfo : Exception);

### *Description*

Writes trace information to the trace log, including any user-defined categories, trace messages, and error information. The trace category that receives the message. The trace message to write to the log. An **System.Exception** that contains information about the error.

TraceMode enumeration (System.Web)

Write

### *Description*

Specifies in what order trace messages are emitted into the HTML output of a page.

Write

[C#] public const TraceMode Default;

1	[C++]	public:	const	TraceMode	Default;
2	[VB]	Public	Const	Default	As TraceMode
3	[JScript]	public	var	Default	: TraceMode;

#### *Description*

Specifies the default value of the **TraceMode** enumeration, which is **SortByTime**.

### **System.Web.Caching**

#### *Description*

The System.Web.Caching namespace provides classes for caching frequently used resources on the server. This includes ASP.NET pages, web services, and user controls. Additionally, a cache dictionary is available for you to store frequently used resources, such as hashtables and other data structures.

Cache class (System.Web.Caching)

#### *Description*

Implements the cache for a Web application.

One instance of this class is created per application domain, and it remains valid as long as the application domain remains active. Information about an instance of this class is available through the **Cache** property of the **System.Web.HttpContext** object. This class cannot be inherited.

1  
2 [C#] public static readonly DateTime NoAbsoluteExpiration;  
3 [C++] public: static DateTime NoAbsoluteExpiration;  
4 [VB] Public Shared ReadOnly NoAbsoluteExpiration As DateTime  
5 [JScript] public static var NoAbsoluteExpiration : DateTime;  
6

7 *Description*

8       Used in the *absoluteExpiration* parameter in an  
9 **System.Web.Caching.Cache.Insert(System.String,System.Object)** method call  
10 to indicate the item should never expire. This field is read-only.

11       When used, this field sets the *absoluteExpiration* parameter equal to  
12 **System.DateTime.MaxValue** , which is a constant representing the largest  
13 possible **DateTime** value, 12/31/9999 11:59:59 PM.  
14

15 [C#] public static readonly TimeSpan NoSlidingExpiration;  
16 [C++] public: static TimeSpan NoSlidingExpiration;  
17 [VB] Public Shared ReadOnly NoSlidingExpiration As TimeSpan  
18 [JScript] public static var NoSlidingExpiration : TimeSpan;  
19

20 *Description*

21       Used as the *slidingExpiration* parameter in an  
22 **System.Web.Caching.Cache.Insert(System.String,System.Object)** method call  
23 to disable sliding expirations. This field is read-only.  
24  
25

1 When used, this field sets the *slidingExpiration* parameter equal to the  
2 **System.TimeSpan.Zero** field, which has a constant value of zero. The cached  
3 item then expires in accordance with the *absoluteExpiration* parameter.

4 Constructors:

5 Cache

6 *Example Syntax:*

7  
8 [C#] public Cache();

9 [C++] public: Cache();

10 [VB] Public Sub New()

11 [JScript] public function Cache();

12 Properties:

13 Count

14  
15 [C#] public int Count {get;}

16 [C++] public: \_\_property int get\_Count();

17 [VB] Public ReadOnly Property Count As Integer

18 [JScript] public function get Count() : int;

19  
20 *Description*

21 Gets the number of items stored in the cache.

22 This property can be useful when monitoring your application's  
23 performance or when using ASP.NET tracing functionality.

24 Item

```

1 [C#] public object this[string key] {get; set;}
2
3 [C++] public: __property Object* get_Item(String* key);public: __property void
4 set_Item(String* key, Object*);
5
6 [VB] Public Default Property Item(ByVal key As String) As Object
7
8 [JScript] returnValue = CacheObject.Item(key);CacheObject.Item(key) =
9 returnValue;

```

### *Description*

Gets or sets the cache item at the specified key.

You can use this property to retrieve the value of a specified cache item, or to add an item and a key for it to the cache. A **System.String** object that represents the key for the cache item.

### Methods:

#### Add

```

17 [C#] public object Add(string key, object value, CacheDependency dependencies,
18 DateTime absoluteExpiration, TimeSpan slidingExpiration, CacheItemPriority
19 priority, CacheItemRemovedCallback onRemoveCallback);
20
21 [C++] public: Object* Add(String* key, Object* value, CacheDependency*
22 dependencies, DateTime absoluteExpiration, TimeSpan slidingExpiration,
23 CacheItemPriority priority, CacheItemRemovedCallback* onRemoveCallback);
24
25 [VB] Public Function Add(ByVal key As String, ByVal value As Object, ByVal
dependencies As CacheDependency, ByVal absoluteExpiration As DateTime,
ByVal slidingExpiration As TimeSpan, ByVal priority As CacheItemPriority,

```

1 ByVal onRemoveCallback As CacheItemRemovedCallback) As Object  
2 [JScript] public function Add(key : String, value : Object, dependencies :  
3 CacheDependency, absoluteExpiration : DateTime, slidingExpiration : TimeSpan,  
4 priority : CacheItemPriority, onRemoveCallback : CacheItemRemovedCallback) :  
5 Object;

### 7 *Description*

8 Adds the specified item to the **System.Web.Caching.Cache** object with  
9 dependencies, expiration and priority policies, and a delegate you can use to notify  
10 your application when the inserted item is removed from the **Cache** .

11 *Return Value:* The **System.Object** item added to the **Cache** . The cache key used  
12 to reference the item. The item to be added to the cache. The file or cache key  
13 dependencies for the item. When any dependency changes, the object becomes  
14 invalid and is removed from the cache. If there are no dependencies, this paramter  
15 contains **null** . The time at which the added object expires and is removed from the  
16 cache. The interval between the time the added object was last accessed and when  
17 that object expires. If this value is the equivalent of 20 minutes, the object expires  
18 and is removed from the cache 20 minutes after it is last accessed. The relative  
19 cost of the object, as expressed by the **System.Web.Caching.CacheItemPriority**  
20 enumeration. The cache uses this value when it evicts objects; objects with a lower  
21 cost are removed from the cache before objects with a higher cost. A delegate that,  
22 if provided, is called when an object is removed from the cache. You can use this  
23 to notify applications when their objects are deleted from the cache.

24 Get



1  
2 [C#] public object Get(string key);  
3 [C++] public: Object\* Get(String\* key);  
4 [VB] Public Function Get(ByVal key As String) As Object  
5 [JScript] public function Get(key : String) : Object;  
6

### 7 *Description*

8       Retrieves the specified item from the **System.Web.Caching.Cache** object.

9 *Return Value:* The retrieved cache item, or **null** if the key is not found. The  
10 identifier for the cache item to retrieve.

### 11       GetEnumerator

12  
13 [C#] public IDictionaryEnumerator GetEnumerator();  
14 [C++] public: IDictionaryEnumerator\* GetEnumerator();  
15 [VB] Public Function GetEnumerator() As IDictionaryEnumerator  
16 [JScript] public function GetEnumerator() : IDictionaryEnumerator;  
17

### 18 *Description*

19       Retrieves a dictionary enumerator used to iterate through the key settings  
20 and their values contained in the cache.

21 *Return Value:* An enumerator to iterate through the **System.Web.Caching.Cache**  
22 object.

23       Items can be added or removed from the cache while this method is  
24 enumerating through the items.

### 25       Insert

1  
2 [C#] public void Insert(string key, object value);  
3 [C++] public: void Insert(String\* key, Object\* value);  
4 [VB] Public Sub Insert(ByVal key As String, ByVal value As Object)  
5 [JScript] public function Insert(key : String, value : Object); Inserts an item into  
6 the **System.Web.Caching.Cache** object.

7  
8 *Description*

9 Inserts an item into the **System.Web.Caching.Cache** object with a cache  
10 key to reference its location and using default values provided by the  
11 **System.Web.Caching.CacheItemPriority** and  
12 **System.Web.Caching.CacheItemPriorityDecay** enumerations. The cache key  
13 used to reference the item. The object to be inserted into the cache.

14 **Insert**

15  
16 [C#] public void Insert(string key, object value, CacheDependency dependencies);  
17 [C++] public: void Insert(String\* key, Object\* value, CacheDependency\*  
18 dependencies);  
19 [VB] Public Sub Insert(ByVal key As String, ByVal value As Object, ByVal  
20 dependencies As CacheDependency)  
21 [JScript] public function Insert(key : String, value : Object, dependencies :  
22 CacheDependency);

23  
24 *Description*  
25

1 Inserts an object into the **System.Web.Caching.Cache** that has file or key  
2 dependencies. The cache key used to identify the item. The object to be inserted in  
3 the cache. The file or cache key dependencies for the inserted object. When any  
4 dependency changes, the object becomes invalid and is removed from the cache. If  
5 there are no dependencies, this parameter contains **null**.

## 6 Insert

7  
8 [C#] public void Insert(string key, object value, CacheDependency dependencies,  
9 DateTime absoluteExpiration, TimeSpan slidingExpiration);

10 [C++] public: void Insert(String\* key, Object\* value, CacheDependency\*  
11 dependencies, DateTime absoluteExpiration, TimeSpan slidingExpiration);

12 [VB] Public Sub Insert(ByVal key As String, ByVal value As Object, ByVal  
13 dependencies As CacheDependency, ByVal absoluteExpiration As DateTime,  
14 ByVal slidingExpiration As TimeSpan)

15 [JScript] public function Insert(key : String, value : Object, dependencies :  
16 CacheDependency, absoluteExpiration : DateTime, slidingExpiration :  
17 TimeSpan); .

## 18 19 Description

20 Inserts an object into the **System.Web.Caching.Cache** with dependencies  
21 and expiration policies.

22 If the *slidingExpiration* parameter is set to **NoSlidingExpiration**, sliding  
23 expiration is disabled. If you set the *slidingExpiration* parameter to greater than  
24 **System.TimeSpan.Zero**, the *absoluteExpiration* parameter is set to  
25 **System.DateTime.Now** plus the value contained in the *slidingExpiration*

parameter. If the item is requested from the cache before the amount of time specified by the *absoluteExpiration* parameter, the item will be placed in the cache again, and *absoluteExpiration* will again be set to **DateTime.Now** plus the value contained in the *slidingExpiration* parameter. If the item is not requested from the cache before the date in the *absoluteExpiration* parameter, the item is removed from the cache. The cache key used to reference the object. The object to be inserted in the cache. The file or cache key dependencies for the inserted object. When any dependency changes, the object becomes invalid and is removed from the cache. If there are no dependencies, this parameter contains **null**. The time at which the inserted object expires and is removed from the cache. The interval between the time the inserted object is last accessed and when that object expires. If this value is the equivalent of 20 minutes, the object will expire and be removed from the cache 20 minutes after it was last accessed.

#### Insert

```
[C#] public void Insert(string key, object value, CacheDependency dependencies,
DateTime absoluteExpiration, TimeSpan slidingExpiration, CacheItemPriority
priority, CacheItemRemovedCallback onRemoveCallback);
```

```
[C++] public: void Insert(String* key, Object* value, CacheDependency*
dependencies, DateTime absoluteExpiration, TimeSpan slidingExpiration,
CacheItemPriority priority, CacheItemRemovedCallback* onRemoveCallback);
```

```
[VB] Public Sub Insert(ByVal key As String, ByVal value As Object, ByVal
dependencies As CacheDependency, ByVal absoluteExpiration As DateTime,
ByVal slidingExpiration As TimeSpan, ByVal priority As CacheItemPriority,
ByVal onRemoveCallback As CacheItemRemovedCallback)
```

```
1 [JScript] public function Insert(key : String, value : Object, dependencies :  
2 CacheDependency, absoluteExpiration : DateTime, slidingExpiration : TimeSpan,  
3 priority : CacheItemPriority, onRemoveCallback : CacheItemRemovedCallback);  
4
```

### 5 *Description*

6 Inserts an object into the **System.Web.Caching.Cache** object with  
7 dependencies, expiration and priority policies, and a delegate you can use to notify  
8 your application when the inserted item is removed from the **Cache**. The cache  
9 key used to reference the object. The object to be inserted in the cache. The file or  
10 cache key dependencies for the item. When any dependency changes, the object  
11 becomes invalid and is removed from the cache. If there are no dependencies, this  
12 parameter contains **null**. The time at which the inserted object expires and is  
13 removed from the cache. The interval between the time the inserted object was last  
14 accessed and when that object expires. If this value is the equivalent of 20  
15 minutes, the object will expire and be removed from the cache 20 minutes after it  
16 was last accessed. The cost of the object relative to other items stored in the cache,  
17 as expressed by the **System.Web.Caching.CacheItemPriority** enumeration. This  
18 value is used by the cache when it evicts objects; objects with a lower cost are  
19 removed from the cache before objects with a higher cost. A delegate that, if  
20 provided, will be called when an object is removed from the cache. You can use  
21 this to notify applications when their objects are deleted from the cache.

### 22 **Remove**

```
23  
24 [C#] public object Remove(string key);  
25 [C++] public: Object* Remove(String* key);
```

1 [VB] Public Function Remove(ByVal key As String) As Object

2 [JScript] public function Remove(key : String) : Object;

3  
4 *Description*

5 Removes the specified item from the application's  
6 **System.Web.Caching.Cache** object.

7 *Return Value:* The item removed from the **Cache** . If the value in the key  
8 parameter is not found, returns **null** . A **System.String** identifier for the cache  
9 item to remove.

10 **IEnumerable.GetEnumerator**

11  
12 [C#] IEnumerator IEnumerable.GetEnumerator();

13 [C++] IEnumerator\* IEnumerable::GetEnumerator();

14 [VB] Function GetEnumerator() As IEnumerator Implements  
15 **IEnumerable.GetEnumerator**

16 [JScript] function IEnumerable.GetEnumerator() : IEnumerator;

17 **CacheDependency** class (System.Web.Caching)

18 **ToString**

19  
20  
21 *Description*

22 Tracks cache dependencies, which can be files, directories, or keys to other  
23 objects in your application's **System.Web.Caching.Cache** . This class cannot be  
24 inherited.

1 You can add items with dependencies to your application's cache with the  
2 **System.Web.Caching.Cache.Add(System.String,System.Object,System.Web.**  
3 **Caching.CacheDependency,System.DateTime,System.TimeSpan,System.Web.**  
4 **Caching.CacheItemPriority,System.Web.Caching.CacheItemRemovedCallba**  
5 **ck)** and **System.Web.Caching.Cache.Insert(System.String,System.Object)**  
6 methods.

7 CacheDependency

8 *Example Syntax:*

9 ToString

10  
11 [C#] public CacheDependency(string filename);

12 [C++] public: CacheDependency(String\* filename);

13 [VB] Public Sub New(ByVal filename As String)

14 [JScript] public function CacheDependency(filename : String); Initializes a new  
15 instance of the **System.Web.Caching.CacheDependency** class.

16  
17 *Description*

18 Initializes a new instance of the **System.Web.Caching.CacheDependency**  
19 class that monitors a file or directory for changes. The path to a file or directory  
20 that the cached object is dependent upon. When this resource changes, the cached  
21 object becomes obsolete and is removed from the cache.

22 CacheDependency

23 *Example Syntax:*

24 ToString

```

1
2 [C#] public CacheDependency(string[] filenames);
3 [C++] public: CacheDependency(String* filenames __gc[]);
4 [VB] Public Sub New(ByVal filenames() As String)
5 [JScript] public function CacheDependency(filenames : String[]);
6

```

### 7 *Description*

8        Initializes a new instance of the **System.Web.Caching.CacheDependency**  
9 class that monitors an array of file paths (to files or directories) for changes.

10       If any of the files or directories in the array were to change or be removed  
11 from the array, the cached item becomes obsolete and is removed from the  
12 application's **System.Web.Caching.Cache** object. An array of file paths (to files  
13 or directories) that the cached object is dependent upon. When any of these  
14 resources change, the cached object becomes obsolete and is removed from the  
15 cache.

16       CacheDependency

17       *Example Syntax:*

18       ToString

```

19
20 [C#] public CacheDependency(string filename, DateTime start);
21 [C++] public: CacheDependency(String* filename, DateTime start);
22 [VB] Public Sub New(ByVal filename As String, ByVal start As DateTime)
23 [JScript] public function CacheDependency(filename : String, start : DateTime);
24

```

### 25 *Description*



1        Initializes a new instance of the **System.Web.Caching.CacheDependency**  
2 class that monitors a file or directory for changes and indicates when change  
3 tracking is to begin. The path to a file or directory that the cached object is  
4 dependent upon. When this resource changes, the cached object becomes obsolete  
5 and is removed from the cache. The time when change tracking begins.

6        CacheDependency

7        *Example Syntax:*

8        ToString

9  
10      [C#] public CacheDependency(string[] filenames, DateTime start);  
11      [C++] public: CacheDependency(String\* filenames \_\_gc[], DateTime start);  
12      [VB] Public Sub New(ByVal filenames() As String, ByVal start As DateTime)  
13      [JScript] public function CacheDependency(filenames : String[], start :  
14      DateTime);

15  
16      *Description*

17        Initializes a new instance of the **System.Web.Caching.CacheDependency**  
18 class that monitors an array of file paths (to files or directories) for changes and  
19 specifies a time when change monitoring begins. An array of file paths (to files or  
20 directories) that the cached object is dependent upon. When any of these resources  
21 change, the cached object becomes obsolete and is removed from the cache. The  
22 time when change tracking begins.

23        CacheDependency

24        *Example Syntax:*

25        ToString

```

1
2 [C#] public CacheDependency(string[] filenames, string[] cachekeys);
3 [C++] public: CacheDependency(String* filenames __gc[], String* cachekeys
4 __gc[]);
5 [VB] Public Sub New(ByVal filenames() As String, ByVal cachekeys() As String)
6 [JScript] public function CacheDependency(filenames : String[], cachekeys :
7 String[]);
8

```

### 9 *Description*

10        Initializes a new instance of the **System.Web.Caching.CacheDependency**  
11 class that monitors an array of file paths (to files or directories), an array of cache  
12 keys, or both for changes. An array of file paths (to files or directories) that the  
13 cached object is dependent upon. When any of these resources change, the cached  
14 object becomes obsolete and is removed from the cache. An array of cache keys  
15 that the new object monitors for changes. When any of these cache keys change,  
16 the cached object associated with this dependency object becomes obsolete and is  
17 removed from the cache.

18        CacheDependency

19        *Example Syntax:*

20        ToString

```

21
22 [C#] public CacheDependency(string[] filenames, string[] cachekeys,
23 CacheDependency dependency);
24 [C++] public: CacheDependency(String* filenames __gc[], String* cachekeys
25 __gc[], CacheDependency* dependency);

```

```

1 [VB] Public Sub New(ByVal filenames() As String, ByVal cachekeys() As String,
2   ByVal dependency As CacheDependency)
3 [JScript] public function CacheDependency(filenames : String[], cachekeys :
4   String[], dependency : CacheDependency);
5

```

### *Description*

Initializes a new instance of the **System.Web.Caching.CacheDependency** class that monitors an array of file paths (to files or directories), an array of cache keys, or both for changes. It also makes itself dependent upon a separate instance of the **CacheDependency** class. An array of file paths (to files or directories) that the cached object is dependent upon. When any of these resources change, the cached object becomes obsolete and is removed from the cache. An array of cache keys that the new object monitors for changes. When any of these cache keys change, the cached object associated with this dependency object becomes obsolete and is removed from the cache. Another instance of the **CacheDependency** class that this instance is dependent upon.

CacheDependency

*Example Syntax:*

ToString

```

21 [C#] public CacheDependency(string[] filenames, string[] cachekeys, DateTime
22   start);
23 [C++] public: CacheDependency(String* filenames __gc[], String* cachekeys
24   __gc[], DateTime start);
25 [VB] Public Sub New(ByVal filenames() As String, ByVal cachekeys() As String,

```

1 ByVal start As DateTime)

2 [JScript] public function CacheDependency(filenamees : String[], cachekeys :  
3 String[], start : DateTime);

4  
5 *Description*

6       Initializes a new instance of the **System.Web.Caching.CacheDependency**  
7 class that monitors an array of file paths (to files or directories), an array of cache  
8 keys, or both for changes. An array of file paths (to files or directories) that the  
9 cached object is dependent upon. When any of these resources change, the cached  
10 object becomes obsolete and is removed from the cache. An array of cache keys  
11 that the new object monitors for changes. When any of these cache keys change,  
12 the cached object associated with this dependency object becomes obsolete and is  
13 removed from the cache. The time when change tracking begins.

14       CacheDependency

15       *Example Syntax:*

16       ToString

17  
18 [C#] public CacheDependency(string[] filenames, string[] cachekeys,

19 CacheDependency dependency, DateTime start);

20 [C++] public: CacheDependency(String\* filenames \_\_gc[], String\* cachekeys  
21 \_\_gc[], CacheDependency\* dependency, DateTime start);

22 [VB] Public Sub New(ByVal filenames() As String, ByVal cachekeys() As String,  
23 ByVal dependency As CacheDependency, ByVal start As DateTime)

24 [JScript] public function CacheDependency(filenamees : String[], cachekeys :  
25 String[], dependency : CacheDependency, start : DateTime);

## *Description*

Initializes a new instance of the **System.Web.Caching.CacheDependency** class that monitors an array of file paths (to files or directories), an array of cache keys, or both for changes. It also makes itself dependent upon another instance of the **CacheDependency** class and a time when the change monitoring begins. An array of file paths (to files or directories) that the cached object is dependent upon. When any of these resources change, the cached object becomes obsolete and is removed from the cache. An array of cache keys that the new object monitors for changes. When any of these cache keys change, the cached object associated with this dependency object becomes obsolete and is removed from the cache. Another instance of the **CacheDependency** class that this instance is dependent upon. The time when change tracking begins.

## *Dispose*

```
[C#] public void Dispose();  
[C++] public: __sealed void Dispose();  
[VB] NotOverridable Public Sub Dispose()  
[JScript] public function Dispose();
```

## *Description*

Releases the resources used by the **System.Web.Caching.CacheDependency** object.

CacheItemPriority enumeration (System.Web.Caching)

ToString

### *Description*

Specifies the relative priority of items stored in the **System.Web.Caching.Cache**.

When the Web server hosting an ASP.NET application runs low on memory, the **Cache** selectively purges items to free system memory. When an item is added to the cache, you can assign it a relative priority compared to the other items stored in the cache. Items you assign higher priority values to are less likely to be deleted from the cache when the server is processing a large number of requests, while items you assign lower priority values are more likely to be deleted. The default is **Normal**.

### *ToString*

```
[C#] public const CacheItemPriority AboveNormal;  
[C++] public: const CacheItemPriority AboveNormal;  
[VB] Public Const AboveNormal As CacheItemPriority  
[JScript] public var AboveNormal : CacheItemPriority;
```

### *Description*

Cache items with this priority level are less likely to be deleted as the server frees system memory than those assigned a **Normal** priority.

### *ToString*

```
[C#] public const CacheItemPriority BelowNormal;
```

1 [C++] public: const CacheItemPriority BelowNormal;

2 [VB] Public Const BelowNormal As CacheItemPriority

3 [JScript] public var BelowNormal : CacheItemPriority;

4  
5 *Description*

6 Cache items with this priority level are more likely to be deleted from the  
7 cache as the server frees system memory than items assigned a **Normal** priority.

8 ToString

9  
10 [C#] public const CacheItemPriority Default;

11 [C++] public: const CacheItemPriority Default;

12 [VB] Public Const Default As CacheItemPriority

13 [JScript] public var Default : CacheItemPriority;

14  
15 *Description*

16 The default value for a cached item's priority is **Normal** .

17 ToString

18  
19 [C#] public const CacheItemPriority High;

20 [C++] public: const CacheItemPriority High;

21 [VB] Public Const High As CacheItemPriority

22 [JScript] public var High : CacheItemPriority;

23  
24 *Description*

25

1       Cache items with this priority level are the least likely to be deleted from  
2 the cache as the server frees system memory.

3       ToString

4  
5 [C#] public const CacheItemPriority Low;

6 [C++] public: const CacheItemPriority Low;

7 [VB] Public Const Low As CacheItemPriority

8 [JScript] public var Low : CacheItemPriority;

9  
10 *Description*

11       Cache items with this priority level are the most likely to be deleted from  
12 the cache as the server frees system memory.

13       ToString

14  
15 [C#] public const CacheItemPriority Normal;

16 [C++] public: const CacheItemPriority Normal;

17 [VB] Public Const Normal As CacheItemPriority

18 [JScript] public var Normal : CacheItemPriority;

19  
20 *Description*

21       Cache items with this priority level are likely to be deleted from the cache  
22 as the server frees system memory only after those items with **Low** or  
23 **BelowNormal** priority. This is the default.

24       ToString



1  
2 [C#] public const CacheItemPriority NotRemovable;  
3 [C++] public: const CacheItemPriority NotRemovable;  
4 [VB] Public Const NotRemovable As CacheItemPriority  
5 [JScript] public var NotRemovable : CacheItemPriority;  
6

7 *Description*

8 The cache items with this priority level will not be deleted from the cache  
9 as the server frees system memory.

10 CacheItemRemovedCallback delegate (System.Web.Caching)

11 ToString  
12  
13

14 *Description*

15 Defines a callback method for notifying applications when a cached item is  
16 removed from the **System.Web.Caching.Cache** . The index location for the item  
17 removed from the cache. The **System.Object** item removed from the cache. The  
18 reason the item was removed from the cache, as specified by the  
19 **System.Web.Caching.CacheItemRemovedReason** enumeration.

20 CacheItemRemovedReason enumeration (System.Web.Caching)

21 ToString  
22  
23

24 *Description*  
25

1 Specifies the reason an item was removed from the  
2 **System.Web.Caching.Cache** .

3 This enumeration works in concert with the  
4 **System.Web.Caching.CacheItemRemovedCallback** delegate to notify your  
5 ASP.NET applications when and why an object was removed from the  
6 **System.Web.Caching.Cache** .

7 ToString

8  
9 [C#] public const CacheItemRemovedReason DependencyChanged;

10 [C++] public: const CacheItemRemovedReason DependencyChanged;

11 [VB] Public Const DependencyChanged As CacheItemRemovedReason

12 [JScript] public var DependencyChanged : CacheItemRemovedReason;

13  
14 *Description*

15 The item is removed from the cache because a file or key dependency  
16 changed.

17 ToString

18  
19 [C#] public const CacheItemRemovedReason Expired;

20 [C++] public: const CacheItemRemovedReason Expired;

21 [VB] Public Const Expired As CacheItemRemovedReason

22 [JScript] public var Expired : CacheItemRemovedReason;

23  
24 *Description*

25 The item is removed from the cache because it expired.

1 ToString

2  
3 [C#] public const CacheItemRemovedReason Removed;  
4 [C++] public: const CacheItemRemovedReason Removed;  
5 [VB] Public Const Removed As CacheItemRemovedReason  
6 [JScript] public var Removed : CacheItemRemovedReason;  
7

8 *Description*

9 The item is removed fro  
10  
11

12 **System.Web.Configuration**

13  
14 *Description*

15 The System.Web.Configuration namespace contains classes that are used to  
16 set up ASP.NET configuration.

17 AuthenticationMode enumeration (System.Web.Configuration)  
18  
19

20 *Description*

21 Provides enumerated values that specify the authentication mode for an  
22 application.  
23

24 [C#] public const AuthenticationMode Forms;  
25 [C++] public: const AuthenticationMode Forms;

1 [VB] Public Const Forms As AuthenticationMode

2 [JScript] public var Forms : AuthenticationMode;

3  
4 *Description*

5 Specifies ASP.NET forms-based authentication as the default  
6 authentication mode .

7  
8 [C#] public const AuthenticationMode None;

9 [C++] public: const AuthenticationMode None;

10 [VB] Public Const None As AuthenticationMode

11 [JScript] public var None : AuthenticationMode;

12  
13 *Description*

14 Specifies no authentication. Only anonymous users are expected. Also,  
15 applications can handle events to provide their own authentication.

16  
17 [C#] public const AuthenticationMode Passport;

18 [C++] public: const AuthenticationMode Passport;

19 [VB] Public Const Passport As AuthenticationMode

20 [JScript] public var Passport : AuthenticationMode;

21  
22 *Description*

23 Specifies Microsoft Passport authentication as the default authentication  
24 mode.

1  
2 [C#] public const AuthenticationMode Windows;

3 [C++] public: const AuthenticationMode Windows;

4 [VB] Public Const Windows As AuthenticationMode

5 [JScript] public var Windows : AuthenticationMode;

6  
7 *Description*

8 Specifies Windows authentication as the default authentication mode. Use  
9 this mode when using any form of Internet Information Services (IIS)  
10 authentication: Basic, Digest, Integrated Windows authentication  
11 (NTLM/Kerberos), or certificates.

12 *Methods:*

13 ClientTargetSectionHandler class (System.Web.Configuration)

14 ToString

15  
16  
17 *Description*

18  
19 *Constructors:*

20 ClientTargetSectionHandler

21 *Example Syntax:*

22 ToString

23  
24 [C#] public ClientTargetSectionHandler();

25 [C++] public: ClientTargetSectionHandler();

1 [VB] Public Sub New()

2 [JScript] public function ClientTargetSectionHandler();

3 Properties:

4 KeyAttributeName

5 ToString

6  
7 [C#] protected override string KeyAttributeName {get;}

8 [C++] protected: \_\_property virtual String\* get\_KeyAttributeName();

9 [VB] Overrides Protected ReadOnly Property KeyAttributeName As String

10 [JScript] protected function get KeyAttributeName() : String;

11  
12 *Description*

13 Gets the name of the key attribute tag. This property is overridden by  
14 derived classes to change the name of the key attribute tag. The default is "key".

15 ValueAttributeName

16 ToString

17  
18 [C#] protected override string ValueAttributeName {get;}

19 [C++] protected: \_\_property virtual String\* get\_ValueAttributeName();

20 [VB] Overrides Protected ReadOnly Property ValueAttributeName As String

21 [JScript] protected function get ValueAttributeName() : String;

22  
23 *Description*

24 Gets the name of the value tag. This property may be overridden by derived  
25 classes to change the name of the value tag. The default is "value".

FormsAuthPasswordFormat enumeration (System.Web.Configuration)

ToString

*Description*

Provides enumerated values that specify the encryption format for storing passwords.

ToString

[C#] public const FormsAuthPasswordFormat Clear;

[C++] public: const FormsAuthPasswordFormat Clear;

[VB] Public Const Clear As FormsAuthPasswordFormat

[JScript] public var Clear : FormsAuthPasswordFormat;

*Description*

Specifies that passwords are not encrypted.

ToString

[C#] public const FormsAuthPasswordFormat MD5;

[C++] public: const FormsAuthPasswordFormat MD5;

[VB] Public Const MD5 As FormsAuthPasswordFormat

[JScript] public var MD5 : FormsAuthPasswordFormat;

*Description*

Specifies that passwords are encrypted using the MD5 hash algorithm.

ToString

[C#] public const FormsAuthPasswordFormat SHA1;

[C++] public: const FormsAuthPasswordFormat SHA1;

[VB] Public Const SHA1 As FormsAuthPasswordFormat

[JScript] public var SHA1 : FormsAuthPasswordFormat;

*Description*

Specifies that passwords are encrypted using the SHA1 hash algorithm.

FormsProtectionEnum enumeration (System.Web.Configuration)

ToString

*Description*

Provides enumerated values that are used to specify the forms protection method.

ToString

[C#] public const FormsProtectionEnum All;

[C++] public: const FormsProtectionEnum All;

[VB] Public Const All As FormsProtectionEnum

[JScript] public var All : FormsProtectionEnum;

*Description*



1 Specifies that the application uses both data validation and encryption to  
2 protect the cookie. This option uses the configured data validation algorithm  
3 (based on the element). Triple-DES (3DES) is used for encryption, if it is available  
4 and if the key is at least 48 bytes long.

5 ToString

6  
7 [C#] public const FormsProtectionEnum Encryption;  
8 [C++] public: const FormsProtectionEnum Encryption;  
9 [VB] Public Const Encryption As FormsProtectionEnum  
10 [JScript] public var Encryption : FormsProtectionEnum;

11  
12 *Description*

13 Specifies that the cookie is encrypted using Triple-DES or DES, but data  
14 validation is not performed on the cookie. Cookies used this way might be subject  
15 to plain-text security attacks.

16 ToString

17  
18 [C#] public const FormsProtectionEnum None;  
19 [C++] public: const FormsProtectionEnum None;  
20 [VB] Public Const None As FormsProtectionEnum  
21 [JScript] public var None : FormsProtectionEnum;

22  
23 *Description*

24 Specifies that both encryption and validation are disabled for sites that are  
25 using cookies only for personalization and have weaker security requirements.

1 Using cookies in this manner is not recommended; however, it is the least  
2 resource-intensive way to enable personalization using the .NET Framework.

3 ToString

4  
5 [C#] public const FormsProtectionEnum Validation;

6 [C++] public: const FormsProtectionEnum Validation;

7 [VB] Public Const Validation As FormsProtectionEnum

8 [JScript] public var Validation : FormsProtectionEnum;

9  
10 *Description*

11 Specifies that the application uses a validation scheme to verify that the  
12 contents of an encrypted cookie have not been altered in transit. The cookie is  
13 created by concatenating a validation key with the cookie data, computing a  
14 Message Authentication Code (MAC), and appending the MAC to the outgoing  
15 cookie.

16 HttpCapabilitiesBase class (System.Web.Configuration)

17 ToString

18  
19  
20 *Description*

21 Defines the base class for client browser capabilities.

22 HttpCapabilitiesBase

23 *Example Syntax:*

24 ToString

```

1
2 [C#] public HttpCapabilitiesBase();
3 [C++] public: HttpCapabilitiesBase();
4 [VB] Public Sub New()
5 [JScript] public function HttpCapabilitiesBase();
6     Item
7     ToString
8
9 [C#] public virtual string this[string key] {get;}
10 [C++] public: __property virtual String* get_Item(String* key);
11 [VB] Overridable Public Default ReadOnly Property Item(ByVal key As String)
12 As String
13 [JScript] returnValue = HttpCapabilitiesBaseObject.Item(key);
14
15 Description
16     Allows access to individual dictionary values. The name of the dictionary
17 value to retrieve.
18     GetConfigCapabilities
19
20 [C#] public static HttpCapabilitiesBase GetConfigCapabilities(string configKey,
21 HttpRequest request);
22 [C++] public: static HttpCapabilitiesBase* GetConfigCapabilities(String*
23 configKey, HttpRequest* request);
24 [VB] Public Shared Function GetConfigCapabilities(ByVal configKey As String,
25 ByVal request As HttpRequest) As HttpCapabilitiesBase

```

1 [JScript] public static function GetConfigCapabilities(configKey : String, request :  
2 HttpRequest) : HttpCapabilitiesBase;

3  
4 *Description*

5 Returns individual browser capabilities for the current request. The name of  
6 the requested browser capability. The current **System.Web.HttpContext.Request**

7  
8 **Init**

9  
10 [C#] protected virtual void Init();

11 [C++] protected: virtual void Init();

12 [VB] Overridable Protected Sub Init()  
13  
14  
15

16 **System.Web.Hosting**

17  
18 *Description*

19  
20 AppDomainFactory class (System.Web.Hosting)  
21  
22

23 *Description*

24  
25 **Constructors:**

## AppDomainFactory

### *Example Syntax:*

```
[C#] public AppDomainFactory();
```

```
[C++] public: AppDomainFactory();
```

```
[VB] Public Sub New()
```

```
[JScript] public function AppDomainFactory();
```

### Methods:

#### Create

```
[C#] public object Create(string module, string typeName, string appId, string  
appPath, string strUrlOfAppOrigin, int iZone);
```

```
[C++] public: __sealed Object* Create(String* module, String* typeName,  
String* appId, String* appPath, String* strUrlOfAppOrigin, int iZone);
```

```
[VB] NotOverridable Public Function Create(ByVal module As String, ByVal  
typeName As String, ByVal appId As String, ByVal appPath As String, ByVal  
strUrlOfAppOrigin As String, ByVal iZone As Integer) As Object
```

```
[JScript] public function Create(module : String, typeName : String, appId :  
String, appPath : String, strUrlOfAppOrigin : String, iZone : int) : Object;
```

### *Description*

ApplicationHost class (System.Web.Hosting)

ToString

1  
2  
3 *Description*  
4

5 CreateApplicationHost  
6

7 [C#] public static object CreateApplicationHost(Type hostType, string virtualDir,  
8 string physicalDir);

9 [C++] public: static Object\* CreateApplicationHost(Type\* hostType, String\*  
10 virtualDir, String\* physicalDir);

11 [VB] Public Shared Function CreateApplicationHost(ByVal hostType As Type,  
12 ByVal virtualDir As String, ByVal physicalDir As String) As Object

13 [JScript] public static function CreateApplicationHost(hostType : Type, virtualDir  
14 : String, physicalDir : String) : Object;  
15

16 *Description*  
17

18 IAppDomainFactory interface (System.Web.Hosting)

19 ToString  
20  
21

22 *Description*  
23

24 Create  
25

[C#] object Create(in string module, in string typeName, in string appId, in string appPath, in string strUrlOfAppOrigin, in int iZone);

[C++] Object\* Create(\_\_in String\* module, \_\_in String\* typeName, \_\_in String\* appId, \_\_in String\* appPath, \_\_in String\* strUrlOfAppOrigin, \_\_in int iZone);

[VB] Function Create(ByVal module As String, ByVal typeName As String, ByVal appId As String, ByVal appPath As String, ByVal strUrlOfAppOrigin As String, ByVal iZone As Integer) As Object

[JScript] function Create(in module : String, in typeName : String, in appId : String, in appPath : String, in strUrlOfAppOrigin : String, in iZone : int) : Object;

### *Description*

IISAPIRuntime interface (System.Web.Hosting)

Create

### *Description*

DoGCCollect

[C#] void DoGCCollect();

[C++] void DoGCCollect();

[VB] Sub DoGCCollect()

[JScript] function DoGCCollect();

1  
2 *Description*

3  
4 ProcessRequest

5  
6 [C#] int ProcessRequest(in IntPtr ecb, in int useProcessModel);  
7 [C++] int ProcessRequest(\_\_in IntPtr ecb, \_\_in int useProcessModel);  
8 [VB] Function ProcessRequest(ByVal ecb As IntPtr, ByVal useProcessModel As  
9 Integer) As Integer  
10 [JScript] function ProcessRequest(in ecb : IntPtr, in useProcessModel : int) : int;

11  
12 *Description*

13  
14 StartProcessing

15  
16 [C#] void StartProcessing();  
17 [C++] void StartProcessing();  
18 [VB] Sub StartProcessing()  
19 [JScript] function StartProcessing();

20  
21 *Description*

22  
23 StopProcessing

24  
25 [C#] void StopProcessing();



1 [C++] void StopProcessing();  
2 [VB] Sub StopProcessing()  
3 [JScript] function StopProcessing();  
4

5 *Description*

6  
7 ISAPIRuntime class (System.Web.Hosting)

8 StopProcessing  
9

10  
11 *Description*

12  
13 ISAPIRuntime

14 *Example Syntax:*

15 StopProcessing  
16

17 [C#] public ISAPIRuntime();

18 [C++] public: ISAPIRuntime();

19 [VB] Public Sub New()

20 [JScript] public function ISAPIRuntime();

21 DoGCCollect  
22

23 [C#] public void DoGCCollect();

24 [C++] public: \_\_sealed void DoGCCollect();

25 [VB] NotOverridable Public Sub DoGCCollect()

1 [JScript] public function DoGCCollect();

2  
3 *Description*

4  
5 ProcessRequest

6  
7 [C#] public int ProcessRequest(IntPtr ecb, int iWRTType);

8 [C++] public: \_\_sealed int ProcessRequest(IntPtr ecb, int iWRTType);

9 [VB] NotOverridable Public Function ProcessRequest(ByVal ecb As IntPtr,  
10 ByVal iWRTType As Integer) As Integer

11 [JScript] public function ProcessRequest(ecb : IntPtr, iWRTType : int) : int;

12  
13 *Description*

14  
15 StartProcessing

16  
17 [C#] public void StartProcessing();

18 [C++] public: \_\_sealed void StartProcessing();

19 [VB] NotOverridable Public Sub StartProcessing()

20 [JScript] public function StartProcessing();

21  
22 *Description*

23  
24 StopProcessing

1  
2 [C#] public void StopProcessing();  
3 [C++] public: \_\_sealed void StopProcessing();  
4 [VB] NotOverridable Public Sub StopProcessing()  
5 [JScript] public function StopProcessing();  
6

7 *Description*

8  
9 SimpleWorkerRequest class (System.Web.Hosting)

10 ToString  
11  
12

13 *Description*

14  
15 SimpleWorkerRequest

16 *Example Syntax:*

17 ToString  
18

19 [C#] public SimpleWorkerRequest(string page, string query, TextWriter output);  
20 [C++] public: SimpleWorkerRequest(String\* page, String\* query, TextWriter\*  
21 output);  
22 [VB] Public Sub New(ByVal page As String, ByVal query As String, ByVal  
23 output As TextWriter)  
24 [JScript] public function SimpleWorkerRequest(page : String, query : String,  
25 output : TextWriter);

1  
2 *Description*

3  
4 SimpleWorkerRequest

5 *Example Syntax:*

6 ToString

7  
8 [C#] public SimpleWorkerRequest(string appVirtualDir, string appPhysicalDir,  
9 string page, string query, TextWriter output);

10 [C++] public: SimpleWorkerRequest(String\* appVirtualDir, String\*  
11 appPhysicalDir, String\* page, String\* query, TextWriter\* output);

12 [VB] Public Sub New(ByVal appVirtualDir As String, ByVal appPhysicalDir As  
13 String, ByVal page As String, ByVal query As String, ByVal output As  
14 TextWriter)

15 [JScript] public function SimpleWorkerRequest(appVirtualDir : String,  
16 appPhysicalDir : String, page : String, query : String, output : TextWriter);  
17

18 *Description*

19  
20 Properties:

21 MachineConfigPath

22 ToString

23  
24 [C#] public override string MachineConfigPath {get;}

25 [C++] public: \_\_property virtual String\* get\_MachineConfigPath();

1 [VB] Overrides Public ReadOnly Property MachineConfigPath As String

2 [JScript] public function get MachineConfigPath() : String;

3  
4 *Description*

5  
6 MachineInstallDirectory

7 ToString

8  
9 [C#] public override string MachineInstallDirectory {get;}

10 [C++] public: \_\_property virtual String\* get\_MachineInstallDirectory();

11 [VB] Overrides Public ReadOnly Property MachineInstallDirectory As String

12 [JScript] public function get MachineInstallDirectory() : String;

13  
14 *Description*

15  
16 EndOfRequest

17  
18 [C#] public override void EndOfRequest();

19 [C++] public: void EndOfRequest();

20 [VB] Overrides Public Sub EndOfRequest()

21 [JScript] public override function EndOfRequest();

22  
23 *Description*

24  
25 FlushResponse

1  
2 [C#] public override void FlushResponse(bool finalFlush);  
3 [C++] public: void FlushResponse(bool finalFlush);  
4 [VB] Overrides Public Sub FlushResponse(ByVal finalFlush As Boolean)  
5 [JScript] public override function FlushResponse(finalFlush : Boolean);  
6

7 *Description*  
8

9       GetAppPath  
10

11 [C#] public override string GetAppPath();  
12 [C++] public: String\* GetAppPath();  
13 [VB] Overrides Public Function GetAppPath() As String  
14 [JScript] public override function GetAppPath() : String;  
15

16 *Description*  
17

18       GetAppPathTranslated  
19

20 [C#] public override string GetAppPathTranslated();  
21 [C++] public: String\* GetAppPathTranslated();  
22 [VB] Overrides Public Function GetAppPathTranslated() As String  
23 [JScript] public override function GetAppPathTranslated() : String;  
24

25 *Description*

## GetFilePath

[C#] public override string GetFilePath();  
[C++] public: String\* GetFilePath();  
[VB] Overrides Public Function GetFilePath() As String  
[JScript] public override function GetFilePath() : String;

### *Description*

## GetFilePathTranslated

[C#] public override string GetFilePathTranslated();  
[C++] public: String\* GetFilePathTranslated();  
[VB] Overrides Public Function GetFilePathTranslated() As String  
[JScript] public override function GetFilePathTranslated() : String;

### *Description*

## GetHttpVerbName

[C#] public override string GetHttpVerbName();  
[C++] public: String\* GetHttpVerbName();  
[VB] Overrides Public Function GetHttpVerbName() As String  
[JScript] public override function GetHttpVerbName() : String;

1  
2 *Description*

3  
4 **GetHttpVersion**

5  
6 [C#] public override string GetHttpVersion();

7 [C++] public: String\* GetHttpVersion();

8 [VB] Overrides Public Function GetHttpVersion() As String

9 [JScript] public override function GetHttpVersion() : String;

10  
11 *Description*

12  
13 **GetLocalAddress**

14  
15 [C#] public override string GetLocalAddress();

16 [C++] public: String\* GetLocalAddress();

17 [VB] Overrides Public Function GetLocalAddress() As String

18 [JScript] public override function GetLocalAddress() : String;

19  
20 *Description*

21  
22 **GetLocalPort**

23  
24 [C#] public override int GetLocalPort();

25 [C++] public: int GetLocalPort();



1 [VB] Overrides Public Function GetLocalPort() As Integer  
2 [JScript] public override function GetLocalPort() : int;

3

4 *Description*

5

6 GetPathInfo

7

8 [C#] public override string GetPathInfo();

9 [C++] public: String\* GetPathInfo();

10 [VB] Overrides Public Function GetPathInfo() As String

11 [JScript] public override function GetPathInfo() : String;

12

13 *Description*

14

15 GetQueryString

16

17 [C#] public override string GetQueryString();

18 [C++] public: String\* GetQueryString();

19 [VB] Overrides Public Function GetQueryString() As String

20 [JScript] public override function GetQueryString() : String;

21

22 *Description*

23

24 GetRawUrl

25

1  
2 [C#] public override string GetRawUrl();  
3 [C++] public: String\* GetRawUrl();  
4 [VB] Overrides Public Function GetRawUrl() As String  
5 [JScript] public override function GetRawUrl() : String;  
6

7 *Description*  
8

9 GetRemoteAddress  
10

11 [C#] public override string GetRemoteAddress();  
12 [C++] public: String\* GetRemoteAddress();  
13 [VB] Overrides Public Function GetRemoteAddress() As String  
14 [JScript] public override function GetRemoteAddress() : String;  
15

16 *Description*  
17

18 GetRemotePort  
19

20 [C#] public override int GetRemotePort();  
21 [C++] public: int GetRemotePort();  
22 [VB] Overrides Public Function GetRemotePort() As Integer  
23 [JScript] public override function GetRemotePort() : int;  
24

25 *Description*

## GetServerVariable

[C#] public override string GetServerVariable(string name);  
[C++] public: String\* GetServerVariable(String\* name);  
[VB] Overrides Public Function GetServerVariable(ByVal name As String) As String  
[JScript] public override function GetServerVariable(name : String) : String;

### *Description*

## GetUriPath

[C#] public override string GetUriPath();  
[C++] public: String\* GetUriPath();  
[VB] Overrides Public Function GetUriPath() As String  
[JScript] public override function GetUriPath() : String;

### *Description*

## GetUserToken

[C#] public override IntPtr GetUserToken();  
[C++] public: IntPtr GetUserToken();  
[VB] Overrides Public Function GetUserToken() As IntPtr

1 [JScript] public override function GetUserToken() : IntPtr;

2  
3 *Description*

4  
5 MapPath

6  
7 [C#] public override string MapPath(string path);

8 [C++] public: String\* MapPath(String\* path);

9 [VB] Overrides Public Function MapPath(ByVal path As String) As String

10 [JScript] public override function MapPath(path : String) : String;

11  
12 *Description*

13  
14 SendKnownResponseHeader

15  
16 [C#] public override void SendKnownResponseHeader(int index, string value);

17 [C++] public: void SendKnownResponseHeader(int index, String\* value);

18 [VB] Overrides Public Sub SendKnownResponseHeader(ByVal index As Integer,  
19 ByVal value As String)

20 [JScript] public override function SendKnownResponseHeader(index : int, value :  
21 String);

22  
23 *Description*

24  
25 SendResponseFromFile

```

1
2 [C#] public override void SendResponseFromFile(IntPtr handle, long offset, long
3 length);
4 [C++] public: void SendResponseFromFile(IntPtr handle, __int64 offset, __int64
5 length);
6 [VB] Overrides Public Sub SendResponseFromFile(ByVal handle As IntPtr,
7 ByVal offset As Long, ByVal length As Long)
8 [JScript] public override function SendResponseFromFile(handle : IntPtr, offset :
9 long, length : long);
10

```

# *Description*

## SendResponseFromFile

```

14
15 [C#] public override void SendResponseFromFile(string filename, long offset,
16 long length);
17 [C++] public: void SendResponseFromFile(String* filename, __int64 offset,
18 __int64 length);
19 [VB] Overrides Public Sub SendResponseFromFile(ByVal filename As String,
20 ByVal offset As Long, ByVal length As Long)
21 [JScript] public override function SendResponseFromFile(filename : String, offset
22 : long, length : long);
23

```

# *Description*

## SendResponseFromMemory

[C#] public override void SendResponseFromMemory(byte[] data, int length);

[C++] public: void SendResponseFromMemory(unsigned char data \_\_gc[], int length);

[VB] Overrides Public Sub SendResponseFromMemory(ByVal data() As Byte, ByVal length As Integer)

[JScript] public override function SendResponseFromMemory(data : Byte[], length : int);

### *Description*

## SendStatus

[C#] public override void SendStatus(int statusCode, string statusDescription);

[C++] public: void SendStatus(int statusCode, String\* statusDescription);

[VB] Overrides Public Sub SendStatus(ByVal statusCode As Integer, ByVal statusDescription As String)

[JScript] public override function SendStatus(statusCode : int, statusDescription : String);

## **System.Web.Mail**

### *Description*

1 The System.Web.Mail namespace contains classes that enable you to  
2 construct and send an email attachment using the SMTP mail service built into  
3 Microsoft Windows 2000.

4 MailAttachment class (System.Web.Mail)

5  
6  
7 *Description*

8 Provides properties and methods to construct an email attachment.

9 Constructors:

10 MailAttachment

11 *Example Syntax:*

12  
13 [C#] public MailAttachment(string filename);

14 [C++] public: MailAttachment(String\* filename);

15 [VB] Public Sub New(ByVal filename As String)

16 [JScript] public function MailAttachment(filename : String); Constructs an email  
17 attachment object.

18  
19 *Description*

20 Constructs a **MailAttachment** object and specifies the file name of the  
21 attachment. Sets the **System.Text.Encoding** property to be UUEncode by default.

22 Name of the attached file.

23 MailAttachment

24 *Example Syntax:*

```

1
2 [C#] public MailAttachment(string filename, MailEncoding encoding);
3 [C++] public: MailAttachment(String* filename, MailEncoding encoding);
4 [VB] Public Sub New(ByVal filename As String, ByVal encoding As
5 MailEncoding)
6 [JScript] public function MailAttachment(filename : String, encoding :
7 MailEncoding);
8

```

### 9 *Description*

10 Constructs an email attachment object and specifies the file name and  
 11 encoding of the attachment. Name of the attached file. The type of  
 12 **System.Web.Mail.MailEncoding** used by the attachment.

13 Properties:

14 Encoding

```

15
16 [C#] public MailEncoding Encoding {get;}
17 [C++] public: __property MailEncoding get_Encoding();
18 [VB] Public ReadOnly Property Encoding As MailEncoding
19 [JScript] public function get Encoding() : MailEncoding;
20

```

### 21 *Description*

22 Indicates the type of encoding used to encode the email attachment.

23 Filename

```

24
25 [C#] public string Filename {get;}

```



1 [C++] public: \_\_property String\* get\_Filename();

2 [VB] Public ReadOnly Property Filename As String

3 [JScript] public function get Filename() : String;

4  
5 *Description*

6 Indicates the name of the file to attach to the email.

7 Methods:

8 MailEncoding enumeration (System.Web.Mail)

9 ToString

10  
11  
12 *Description*

13 Provides enumerated values for email encoding.

14 ToString

15  
16 [C#] public const MailEncoding Base64;

17 [C++] public: const MailEncoding Base64;

18 [VB] Public Const Base64 As MailEncoding

19 [JScript] public var Base64 : MailEncoding;

20  
21 *Description*

22 Specifies that the email message uses Base64 encoding.

23 ToString

24  
25 [C#] public const MailEncoding UUEncode;

1 [C++] public: const MailEncoding UUEncode;  
2 [VB] Public Const UUEncode As MailEncoding  
3 [JScript] public var UUEncode : MailEncoding;  
4

5 *Description*

6 Specifies that the email message uses UUEncode encoding.  
7 MailFormat enumeration (System.Web.Mail)  
8 ToString  
9

10  
11 *Description*

12 Provides enumerated values for email format.  
13 ToString  
14

15 [C#] public const MailFormat Html;  
16 [C++] public: const MailFormat Html;  
17 [VB] Public Const Html As MailFormat  
18 [JScript] public var Html : MailFormat;  
19

20 *Description*

21 Specifies that email format is HTML.  
22 ToString  
23

24 [C#] public const MailFormat Text;  
25 [C++] public: const MailFormat Text;

1 [VB] Public Const Text As MailFormat

2 [JScript] public var Text : MailFormat;

3

4 *Description*

5 Specifies that email format is plain text.

6 MailMessage class (System.Web.Mail)

7 ToString

8

9

10 *Description*

11 Provides properties and methods to construct an email message.

12 MailMessage

13 *Example Syntax:*

14 ToString

15

16 [C#] public MailMessage();

17 [C++] public: MailMessage();

18 [VB] Public Sub New()

19 [JScript] public function MailMessage();

20 Attachments

21 ToString

22

23 [C#] public IList Attachments {get;}

24 [C++] public: \_\_property IList\* get\_Attachments();

25 [VB] Public ReadOnly Property Attachments As IList

1 [JScript] public function get Attachments() : IList;

3 *Description*

4 Specifies the list of attachments that is transmitted with the message.

5 Bcc

6 ToString

8 [C#] public string Bcc {get; set;}

9 [C++] public: \_\_property String\* get\_Bcc();public: \_\_property void

10 set\_Bcc(String\*);

11 [VB] Public Property Bcc As String

12 [JScript] public function get Bcc() : String;public function set Bcc(String);

14 *Description*

15 Gets or sets a semicolon-delimited list of email addresses that receive a  
16 Blind Carbon Copy (BCC) copy of the email message.

17 The primary and carbon copy **System.Web.Mail.MailMessage.Cc**  
18 recipients do not see addresses included in the **Bcc** field.

19 Body

20 ToString

22 [C#] public string Body {get; set;}

23 [C++] public: \_\_property String\* get\_Body();public: \_\_property void

24 set\_Body(String\*);

25 [VB] Public Property Body As String

1 [JScript] public function get Body() : String;public function set Body(String);

3 *Description*

4 Gets or sets the body of the email message.

5 BodyEncoding

6 ToString

8 [C#] public Encoding BodyEncoding {get; set;}

9 [C++] public: \_\_property Encoding\* get\_BodyEncoding();public: \_\_property void  
10 set\_BodyEncoding(Encoding\*);

11 [VB] Public Property BodyEncoding As Encoding

12 [JScript] public function get BodyEncoding() : Encoding;public function set  
13 BodyEncoding(Encoding);

15 *Description*

16 Gets or sets the encoding type of the email body.

17 BodyFormat

18 ToString

20 [C#] public MailFormat BodyFormat {get; set;}

21 [C++] public: \_\_property MailFormat get\_BodyFormat();public: \_\_property void  
22 set\_BodyFormat(MailFormat);

23 [VB] Public Property BodyFormat As MailFormat

24 [JScript] public function get BodyFormat() : MailFormat;public function set  
25 BodyFormat(MailFormat);

1  
2 *Description*

3 Gets or sets the content type of the email body.

4 Cc

5 ToString

6  
7 [C#] public string Cc {get; set;}

8 [C++] public: \_\_property String\* get\_Cc();public: \_\_property void  
9 set\_Cc(String\*);

10 [VB] Public Property Cc As String

11 [JScript] public function get Cc() : String;public function set Cc(String);

12  
13 *Description*

14 Gets or sets a semicolon-delimited list of email addresses that receive a  
15 Carbon Copy (CC) of the email message.

16 From

17 ToString

18  
19 [C#] public string From {get; set;}

20 [C++] public: \_\_property String\* get\_From();public: \_\_property void  
21 set\_From(String\*);

22 [VB] Public Property From As String

23 [JScript] public function get From() : String;public function set From(String);

24  
25 *Description*

1 Gets or sets the email address of the sender.

2 Headers

3 ToString

4  
5 [C#] public IDictionary Headers {get;}

6 [C++] public: \_\_property IDictionary\* get\_Headers();

7 [VB] Public ReadOnly Property Headers As IDictionary

8 [JScript] public function get Headers() : IDictionary;

9  
10 *Description*

11 Specifies the custom headers that are transmitted with the email message.

12 Priority

13 ToString

14  
15 [C#] public MailPriority Priority {get; set;}

16 [C++] public: \_\_property MailPriority get\_Priority();public: \_\_property void  
17 set\_Priority(MailPriority);

18 [VB] Public Property Priority As MailPriority

19 [JScript] public function get Priority() : MailPriority;public function set  
20 Priority(MailPriority);

21  
22 *Description*

23 Gets or sets the priority of the email message.

24 Subject

25 ToString

1  
2 [C#] public string Subject {get; set;}

3 [C++] public: \_\_property String\* get\_Subject();public: \_\_property void  
4 set\_Subject(String\*);

5 [VB] Public Property Subject As String

6 [JScript] public function get Subject() : String;public function set Subject(String);

7  
8 *Description*

9 Gets or sets the subject line of the email message.

10 To

11 ToString

12  
13 [C#] public string To {get; set;}

14 [C++] public: \_\_property String\* get\_To();public: \_\_property void  
15 set\_To(String\*);

16 [VB] Public Property To As String

17 [JScript] public function get To() : String;public function set To(String);

18  
19 *Description*

20 Gets or sets the email address of the recipient.

21 UrlContentBase

22 ToString

23  
24 [C#] public string UrlContentBase {get; set;}

25 [C++] public: \_\_property String\* get\_UrlContentBase();public: \_\_property void



1 set\_UrlContentBase(String\*);

2 [VB] Public Property UrlContentBase As String

3 [JScript] public function get UrlContentBase() : String;public function set

4 UrlContentBase(String);

5  
6 *Description*

7 Gets or sets the URL base of all relative URLs used within the HTML  
8 encoded body.

9 UrlContentLocation

10 ToString

11  
12 [C#] public string UrlContentLocation {get; set;}

13 [C++] public: \_\_property String\* get\_UrlContentLocation();public: \_\_property

14 void set\_UrlContentLocation(String\*);

15 [VB] Public Property UrlContentLocation As String

16 [JScript] public function get UrlContentLocation() : String;public function set

17 UrlContentLocation(String);

18  
19 *Description*

20  
21 MailPriority enumeration (System.Web.Mail)

22 ToString

23  
24  
25 *Description*

Provides enumerated values for email priority.

ToString

[C#] public const MailPriority High;

[C++] public: const MailPriority High;

[VB] Public Const High As MailPriority

[JScript] public var High : MailPriority;

*Description*

Specifies that the email message has high priority.

ToString

[C#] public const MailPriority Low;

[C++] public: const MailPriority Low;

[VB] Public Const Low As MailPriority

[JScript] public var Low : MailPriority;

*Description*

Specifies that the email message has low priority.

ToString

[C#] public const MailPriority Normal;

[C++] public: const MailPriority Normal;

[VB] Public Const Normal As MailPriority

[JScript] public var Normal : MailPriority;

1  
2 *Description*

3 Specifies that the email message has normal priority.

4 SmtMail class (System.Web.Mail)

5 ToString

6  
7  
8 *Description*

9 Provides properties and methods to send an email attachment using the  
10 SMTP mail service built into Microsoft Windows 2000.

11 Mail is by default queued on a Windows 2000 system, ensuring that the  
12 calling program does not block network traffic.

13 SmtMail

14 *Example Syntax:*

15 ToString

16  
17 [C#] public SmtMail();

18 [C++] public: SmtMail();

19 [VB] Public Sub New()

20 [JScript] public function SmtMail();

21 SmtServer

22 ToString

23  
24 [C#] public static string SmtServer {get; set;}

25 [C++] public: \_\_property static String\* get\_SmtServer();public: \_\_property static

1 void set\_SmtpServer(String\*);

2 [VB] Public Shared Property SmtpServer As String

3 [JScript] public static function get SmtpServer() : String;public static function set

4 SmtpServer(String);

5  
6 *Description*

7 Gets or sets the name of the SMTP mail server to use to send email  
8 messages.

9 Send

10  
11 [C#] public static void Send(MailMessage message);

12 [C++] public: static void Send(MailMessage\* message);

13 [VB] Public Shared Sub Send(ByVal message As MailMessage)

14 [JScript] public static function Send(message : MailMessage);

15  
16 *Description*

17 Sends a mail message using arguments supplied in the MailMessage  
18 properties. The **System.Web.Mail.MailMessage** object to send.

19 Send

20  
21 [C#] public static void Send(string from, string to, string subject, string  
22 messageText);

23 [C++] public: static void Send(String\* from, String\* to, String\* subject, String\*  
24 messageText);

25 [VB] Public Shared Sub Send(ByVal from As String, ByVal to As String, ByVal

1 subject As String, ByVal messageText As String)

2 [JScript] public static function Send(from : String, to : String, subject : String,  
3 messageText : String); Sends a mail message.

4  
5 *Description*

6 Sends a mail message using the supplied destination arguments. The  
7 address of the email's sender. The address of the email's recipient. The subject line  
8 of the email message.

9  
10  
11  
12 **System.Web.Security**

13 *Description*

14 This namespace contains classes that are used to implement ASP.NET  
15 security in Web server applications.

16 DefaultAuthenticationEventArgs class (System.Web.Security)

17  
18  
19 *Description*

20 Provides a wrapper around the default authentication services. This class  
21 cannot be inherited.

22 For more information about handling events, see .

23 Constructors:

24 DefaultAuthenticationEventArgs

25 *Example Syntax:*

1  
2 [C#] public DefaultAuthenticationEventArgs(HttpContext context);  
3 [C++] public: DefaultAuthenticationEventArgs(HttpContext\* context);  
4 [VB] Public Sub New(ByVal context As HttpContext)  
5 [JScript] public function DefaultAuthenticationEventArgs(context : HttpContext);  
6

7 *Description*

8       Initializes a new instance of the  
9 **System.Web.Security.DefaultAuthenticationEventArgs** class. The context for  
10 the event.

11       Properties:

12       Context  
13

14 [C#] public HttpContext Context {get;}  
15 [C++] public: \_\_property HttpContext\* get\_Context();  
16 [VB] Public ReadOnly Property Context As HttpContext  
17 [JScript] public function get Context() : HttpContext;  
18

19 *Description*

20       The **HttpContext** intrinsic provides access to **Request** , **Response** , and  
21 **User** objects.

22       Methods:

23       DefaultAuthenticationEventHandler delegate (System.Web.Security)

24       ToString  
25

### *Description*

Represents the method that handles the **DefaultAuthentication\_OnAuthenticate** event of a **DefaultAuthenticationModule**. The object that raised the event. A **DefaultAuthenticationEventArgs** object that contains the event data.

When you create a **System.Web.Security.DefaultAuthenticationEventHandler** delegate, you identify the method to handle the event. To associate the event with your **EventHandler**, add an instance of the delegate to the event. The **EventHandler** is called whenever the event occurs, unless you remove the delegate. For more information about **EventHandler** delegates, see .

**DefaultAuthenticationModule** class (System.Web.Security)

**ToString**

### *Description*

Insures that an **Authentication** object is present in the context. This class cannot be inherited.

**DefaultAuthenticationModule**

*Example Syntax:*

**ToString**

```
[C#] public DefaultAuthenticationModule();
```

1 [C++] public: DefaultAuthenticationModule();

2 [VB] Public Sub New()

3 [JScript] public function DefaultAuthenticationModule();

4 ToString

6 [C#] public event DefaultAuthenticationEventHandler Authenticate;

7 [C++] public: \_\_event DefaultAuthenticationEventHandler\* Authenticate;

8 [VB] Public Event Authenticate As DefaultAuthenticationEventHandler

10 *Description*

11 Defines the event raised by the **DefaultAuthentication** module. Used by  
12 the **DefaultAuthentication\_OnAuthenticate** handler, if one exists.

13 When you create a **System.Web.Security.DefaultAuthenticationModule**  
14 delegate, you identify the method to handle the event. To associate the event with  
15 your EventHandler, add an instance of the delegate to the event. The EventHandler  
16 is called whenever the event occurs, unless you remove the delegate. For more  
17 information about EventHandler delegates, see .

18 Dispose

20 [C#] public void Dispose();

21 [C++] public: \_\_sealed void Dispose();

22 [VB] NotOverridable Public Sub Dispose()

23 [JScript] public function Dispose();

25 *Description*



1        Called by the HTTP runtime to Dispose of the module (derived from  
2 **IHttpModule** ).

3        Call **System.Web.Security.DefaultAuthenticationModule.Dispose** when  
4 you are finished using the **System.Web.Security.DefaultAuthenticationModule**  
5 . The **System.Web.Security.DefaultAuthenticationModule.Dispose** method  
6 leaves the **System.Web.Security.DefaultAuthenticationModule** in an unusable  
7 state. After calling  
8 **System.Web.Security.DefaultAuthenticationModule.Dispose** , you must release  
9 all references to the **System.Web.Security.DefaultAuthenticationModule** so the  
10 memory occupied can be reclaimed by garbage collection.

#### 11        Init

12  
13 [C#] public void Init(HttpApplication app);

14 [C++] public: \_\_sealed void Init(HttpApplication\* app);

15 [VB] NotOverridable Public Sub Init(ByVal app As HttpApplication)

16 [JScript] public function Init(app : HttpApplication);  
17

#### 18        *Description*

19        Initializes the module. The **HTTP application** .

20        FileAuthorizationModule class (System.Web.Security)

#### 21        ToString

#### 24        *Description*

1 Verifies that the remote user has NT permissions to access the file  
2 requested. This class cannot be inherited.

3 This module provides authorization services against file system ACLs.  
4 When the Windows authentication module is being used for the application, this  
5 module ensures (if in the pipeline) that the requesting user is allowed read access  
6 to the resource before executing it.

7 FileAuthorizationModule

8 *Example Syntax:*

9 ToString

10  
11 [C#] public FileAuthorizationModule();

12 [C++] public: FileAuthorizationModule();

13 [VB] Public Sub New()

14 [JScript] public function FileAuthorizationModule();

15 Dispose

16  
17 [C#] public void Dispose();

18 [C++] public: \_\_sealed void Dispose();

19 [VB] NotOverridable Public Sub Dispose()

20 [JScript] public function Dispose();

21  
22 *Description*

23 Disposes of the module derived from **IHttpModule** when called by the  
24 **HttpRuntime** .

1 Call **System.Web.Security.FileAuthorizationModule.Dispose** when you  
2 are finished using the **System.Web.Security.FileAuthorizationModule** . The  
3 **System.Web.Security.FileAuthorizationModule.Dispose** method leaves the  
4 **System.Web.Security.FileAuthorizationModule** in an unusable state. After  
5 calling **System.Web.Security.FileAuthorizationModule.Dispose** , you must  
6 release all references to the **System.Web.Security.FileAuthorizationModule** so  
7 the memory occupied can be reclaimed by garbage collection.

#### 8 Init

9  
10 [C#] public void Init(HttpApplication app);  
11 [C++] public: \_\_sealed void Init(HttpApplication\* app);  
12 [VB] NotOverridable Public Sub Init(ByVal app As HttpApplication)  
13 [JScript] public function Init(app : HttpApplication);  
14

#### 15 *Description*

16 Initializes the module. The **HttpApplication** module.

17 FormsAuthentication class (System.Web.Security)

18 ToString

#### 21 *Description*

22 Provides static methods that supply helper utilities for manipulating  
23 authentication tickets. This class cannot be inherited.

24 FormsAuthentication

25 *Example Syntax:*

```

1      ToString
2
3      [C#] public FormsAuthentication();
4      [C++] public: FormsAuthentication();
5      [VB] Public Sub New()
6      [JScript] public function FormsAuthentication();
7          FormsCookieName
8          ToString
9
10     [C#] public static string FormsCookieName {get;}
11     [C++] public: __property static String* get_FormsCookieName();
12     [VB] Public Shared ReadOnly Property FormsCookieName As String
13     [JScript] public static function get FormsCookieName() : String;

```

#### *Description*

Returns the configured cookie name used for the current application.

Returns a **String** .

FormsCookiePath

ToString

```

21     [C#] public static string FormsCookiePath {get;}
22     [C++] public: __property static String* get_FormsCookiePath();
23     [VB] Public Shared ReadOnly Property FormsCookiePath As String
24     [JScript] public static function get FormsCookiePath() : String;

```

1  
2 *Description*

3 Returns the configured cookie path used for the current application.

4 **Authenticate**

5  
6 [C#] public static bool Authenticate(string name, string password);

7 [C++] public: static bool Authenticate(String\* name, String\* password);

8 [VB] Public Shared Function Authenticate(ByVal name As String, ByVal  
9 password As String) As Boolean

10 [JScript] public static function Authenticate(name : String, password : String) :  
11 Boolean;

12  
13 *Description*

14 Attempts to validate the credentials against those contained in the  
15 configured credential store, given the supplied credentials.

16 *Return Value:* Returns **true** if the credentials are valid, otherwise returns **false** .

17 The user name The user password.

18 **Decrypt**

19  
20 [C#] public static FormsAuthenticationTicket Decrypt(string strEncrypted);

21 [C++] public: static FormsAuthenticationTicket\* Decrypt(String\* strEncrypted);

22 [VB] Public Shared Function Decrypt(ByVal strEncrypted As String) As  
23 FormsAuthenticationTicket

24 [JScript] public static function Decrypt(strEncrypted : String) :

25 FormsAuthenticationTicket;

1  
2 *Description*

3 Returns an instance of a **FormsAuthenticationTicket** class, given an  
4 encrypted authentication ticket obtained from an HTTP cookie.

5 *Return Value:* Returns a **FormsAuthenticationTicket** object.

6 **Encrypt**

7  
8 [C#] public static string Encrypt(FormsAuthenticationTicket ticket);

9 [C++] public: static String\* Encrypt(FormsAuthenticationTicket\* ticket);

10 [VB] Public Shared Function Encrypt(ByVal ticket As

11 FormsAuthenticationTicket) As String

12 [JScript] public static function Encrypt(ticket : FormsAuthenticationTicket) :

13 String;

14  
15 *Description*

16 Produces a string containing an encrypted authentication ticket suitable for  
17 use in an HTTP cookie, given a **FormsAuthenticationTicket** .

18 *Return Value:* Returns a **String** containing an encrypted authentication ticket. An  
19 authentication ticket class.

20 **GetAuthCookie**

21  
22 [C#] public static HttpCookie GetAuthCookie(string userName, bool  
23 createPersistentCookie);

24 [C++] public: static HttpCookie\* GetAuthCookie(String\* userName, bool  
25 createPersistentCookie);

1 [VB] Public Shared Function GetAuthCookie(ByVal userName As String, ByVal  
2 createPersistentCookie As Boolean) As HttpCookie  
3 [JScript] public static function GetAuthCookie(userName : String,  
4 createPersistentCookie : Boolean) : HttpCookie; Creates an authentication cookie  
5 for a given user name.

6  
7 *Description*

8 Creates an authentication cookie for a given user name. This does not set  
9 the cookie as part of the outgoing response, so that an application can have more  
10 control over how the cookie is issued.

11 *Return Value:* Returns an **HttpCookie** . Name of the authenticated user. This does  
12 not have to map to a Windows account. Specifies whether or not a durable cookie  
13 (a cookie that is saved across browser sessions) should be issued. Cookie path  
14 defaults to '/'.

15 **GetAuthCookie**

16  
17 [C#] public static HttpCookie GetAuthCookie(string userName, bool  
18 createPersistentCookie, string strCookiePath);

19 [C++] public: static HttpCookie\* GetAuthCookie(String\* userName, bool  
20 createPersistentCookie, String\* strCookiePath);

21 [VB] Public Shared Function GetAuthCookie(ByVal userName As String, ByVal  
22 createPersistentCookie As Boolean, ByVal strCookiePath As String) As  
23 HttpCookie

24 [JScript] public static function GetAuthCookie(userName : String,  
25 createPersistentCookie : Boolean, strCookiePath : String) : HttpCookie;

## *Description*

Creates an authentication cookie for a given user name. This does not set the cookie as part of the outgoing response, so that an application can have more control over how the cookie is issued.

*Return Value:* Returns an **HttpCookie** . Name of the authenticated user. This does not have to map to a Windows account. Specifies whether or not a durable cookie (a cookie that is saved across browser sessions) should be issued. Specifies cookie path.

## **GetRedirectUrl**

[C#] public static string GetRedirectUrl(string userName, bool createPersistentCookie);

[C++] public: static String\* GetRedirectUrl(String\* userName, bool createPersistentCookie);

[VB] Public Shared Function GetRedirectUrl(ByVal userName As String, ByVal createPersistentCookie As Boolean) As String

[JScript] public static function GetRedirectUrl(userName : String, createPersistentCookie : Boolean) : String;

## *Description*

Returns the redirect URL for the original request that caused the redirect to the logon page.

*Return Value:* Returns a **String** .



1 If there is no original URL, Default.aspx is used. If the redirect URL  
2 specifies a different server, this method also returns the authentication ticket as  
3 part of the query string. This method can be used by applications that need to do  
4 the redirect themselves instead of using **RedirectFromLoginPage** . Name of the  
5 authenticated user. This does not have to map to a Windows account. Specifies  
6 whether or not a durable cookie (a cookie that is saved across browser sessions)  
7 should be issued. Cookie path defaults to '/'.

#### 8 HashPasswordForStoringInConfigFile

9  
10 [C#] public static string HashPasswordForStoringInConfigFile(string password,  
11 string passwordFormat);  
12 [C++] public: static String\* HashPasswordForStoringInConfigFile(String\*  
13 password, String\* passwordFormat);  
14 [VB] Public Shared Function HashPasswordForStoringInConfigFile(ByVal  
15 password As String, ByVal passwordFormat As String) As String  
16 [JScript] public static function HashPasswordForStoringInConfigFile(password :  
17 String, passwordFormat : String) : String;

#### 18 *Description*

19  
20 Given a password and a string identifying the hash type, this routine  
21 produces a hash password suitable for storing in a configuration file.

22 *Return Value:* Returns a **String** containing a hashed password.

23 Password algorithms supported are SHA1 and MD5. The password to hash.  
24 The hash algorithm to use. Choices are "sha1" or "md5".

#### 25 Initialize

1  
2 [C#] public static void Initialize();  
3 [C++] public: static void Initialize();  
4 [VB] Public Shared Sub Initialize()  
5 [JScript] public static function Initialize();  
6

7 *Description*

8        Initializes **FormsAuthentication** by reading the configuration and getting  
9 the cookie values and encryption keys for the given application.

10        RedirectFromLoginPage

11  
12 [C#] public static void RedirectFromLoginPage(string userName, bool  
13 createPersistentCookie);  
14 [C++] public: static void RedirectFromLoginPage(String\* userName, bool  
15 createPersistentCookie);  
16 [VB] Public Shared Sub RedirectFromLoginPage(ByVal userName As String,  
17 ByVal createPersistentCookie As Boolean)  
18 [JScript] public static function RedirectFromLoginPage(userName : String,  
19 createPersistentCookie : Boolean); Redirects an authenticated user back to the  
20 originally requested URL.  
21

22 *Description*

23        Redirects an authenticated user back to the originally requested URL.

24        Redirects, based on the contents of the **ReturnURL** key in the query string,  
25 or redirects to Default.aspx if the return key does not exist. It issues an

1 authentication ticket and does a **SetForms** with the ticket, using the appropriately  
2 configured cookie name for the application as part of the redirect response. Name  
3 of the user for cookie authentication purposes. This does not need to map to an  
4 account name and will be used by URL Authorization. Specifies whether or not a  
5 durable cookie (one that is saved across browser sessions) should be issued.

#### 6 RedirectFromLoginPage

7  
8 [C#] public static void RedirectFromLoginPage(string userName, bool  
9 createPersistentCookie, string strCookiePath);

10 [C++] public: static void RedirectFromLoginPage(String\* userName, bool  
11 createPersistentCookie, String\* strCookiePath);

12 [VB] Public Shared Sub RedirectFromLoginPage(ByVal userName As String,  
13 ByVal createPersistentCookie As Boolean, ByVal strCookiePath As String)

14 [JScript] public static function RedirectFromLoginPage(userName : String,  
15 createPersistentCookie : Boolean, strCookiePath : String);

#### 16 17 *Description*

18 Redirects an authenticated user back to the originally requested URL.

19 Redirects, based on the contents of the **ReturnURL** key in the query string,  
20 or redirects to Default.aspx if the return key does not exist. It issues an  
21 authentication ticket and does a **SetForms** with the ticket, using the appropriately  
22 configured cookie name for the application as part of the redirect response. Name  
23 of the user for cookie authentication purposes. This does not need to map to an  
24 account name and will be used by URL Authorization. Specifies whether or not a  
25

1 durable cookie (one that is saved across browser sessions) should be issued.

2 Specifies cookie path.

3       RenewTicketIfOld

4  
5 [C#] public static FormsAuthenticationTicket

6 RenewTicketIfOld(FormsAuthenticationTicket tOld);

7 [C++] public: static FormsAuthenticationTicket\*

8 RenewTicketIfOld(FormsAuthenticationTicket\* tOld);

9 [VB] Public Shared Function RenewTicketIfOld(ByVal tOld As

10 FormsAuthenticationTicket) As FormsAuthenticationTicket

11 [JScript] public static function RenewTicketIfOld(tOld :

12 FormsAuthenticationTicket) : FormsAuthenticationTicket;

13  
14 *Description*

15       Conditionally updates the sliding expiration on a

16 **FormsAuthenticationTicket**.

17 *Return Value:* Returns the updated **FormsAuthenticationTicket**.

18       The ticket is updated with the **IssueDate** set to the current time and the  
19 expiration based on the difference between the previous expiration time and issue  
20 time. The test used to determine whether the ticket should be updated is (  
21 **DateTime.Now** - **IssueDate**) > ( **Expiration** - **DateTime.Now** ).

22       SetAuthCookie

23  
24 [C#] public static void SetAuthCookie(string userName, bool

25 createPersistentCookie);

```

1 [C++] public: static void SetAuthCookie(String* userName, bool
2 createPersistentCookie);
3 [VB] Public Shared Sub SetAuthCookie(ByVal userName As String, ByVal
4 createPersistentCookie As Boolean)
5 [JScript] public static function SetAuthCookie(userName : String,
6 createPersistentCookie : Boolean); Creates an authentication ticket and attaches it
7 to the cookie's collection of the outgoing response. It does not perform a redirect.

```

### 9 *Description*

10 Creates an authentication ticket for the given *userName* and  
11 *createPersistentCookie* and attaches it to the cookie's collection of the outgoing  
12 response. It does not perform a redirect.

13 This has the effect of creating an authenticated user. Applications can use  
14 this when they intend to manually redirect, or if they do not want to redirect. For  
15 example, you can use this for an embedded logon area on a page. The name of an  
16 authenticated user. This does not have to map to a Windows account. Specifies  
17 whether or not a durable cookie (one that is saved across browser sessions) should  
18 be issued.

### 19 **SetAuthCookie**

```

20
21 [C#] public static void SetAuthCookie(string userName, bool
22 createPersistentCookie, string strCookiePath);
23 [C++] public: static void SetAuthCookie(String* userName, bool
24 createPersistentCookie, String* strCookiePath);
25 [VB] Public Shared Sub SetAuthCookie(ByVal userName As String, ByVal

```

1 createPersistentCookie As Boolean, ByVal strCookiePath As String)

2 [JScript] public static function SetAuthCookie(userName : String,  
3 createPersistentCookie : Boolean, strCookiePath : String);

#### 4 5 *Description*

6 Creates an authentication ticket for the given  
7 *userName*, *createPersistentCookie*, and *strCookiePath* and attaches it to the  
8 cookie's collection of the outgoing response. It does not perform a redirect.

9 This has the effect of creating an authenticated user. Applications can use  
10 this when they intend to manually redirect, or if they do not want to redirect. For  
11 example, you can use this for an embedded logon area on a page. The name of an  
12 authenticated user. This does not have to map to a Windows account. Specifies  
13 whether or not a durable cookie (one that is saved across browser sessions) should  
14 be issued. Specifies cookie path.

#### 15 **SignOut**

16  
17 [C#] public static void SignOut();

18 [C++] public: static void SignOut();

19 [VB] Public Shared Sub SignOut()

20 [JScript] public static function SignOut();

#### 21 22 *Description*

23 Removes the authentication ticket by doing a **SetForms** with an empty  
24 value, given an authenticated user. This removes either durable or session cookies.

25 FormsAuthenticationEventArgs class (System.Web.Security)

ToString

*Description*

Provides data for the **FormsAuthentication\_OnAuthenticate** event. This class cannot be inherited.

For more information about handling events, see .

**FormsAuthenticationEventArgs**

*Example Syntax:*

ToString

[C#] public FormsAuthenticationEventArgs(HttpContext context);

[C++] public: FormsAuthenticationEventArgs(HttpContext\* context);

[VB] Public Sub New(ByVal context As HttpContext)

[JScript] public function FormsAuthenticationEventArgs(context : HttpContext);

*Description*

Initializes a new instance of the **System.Web.Security.FormsAuthenticationEventArgs** class.

The following table shows initial property values for an instance of **System.Web.Security.FormsAuthenticationEventArgs** . The context for the event.

Context

ToString

```

1
2 [C#] public HttpContext Context {get;}
3 [C++] public: __property HttpContext* get_Context();
4 [VB] Public ReadOnly Property Context As HttpContext
5 [JScript] public function get Context() : HttpContext;
6

```

### *Description*

Provides access to **Request** , **Response** , and **User** objects. This is the **HttpContext** intrinsic.

See the ASP.NET **HttpRuntime** documentation for more information.

User

ToString

```

13
14 [C#] public IPrincipal User {get; set;}
15 [C++] public: __property IPrincipal* get_User();public: __property void
16 set_User(IPrincipal*);
17 [VB] Public Property User As IPrincipal
18 [JScript] public function get User() : IPrincipal;public function set
19 User(IPrincipal);
20

```

### *Description*

Indicates the **IPrincipal** object to be associated with the request.

The user object will be attached to the context. If **User** is non null and **Context.User** is null, the **FormsAuthenticationModule** will initialize **Context.User** with **FormsAuthenticationEventArgs.User** .



FormsAuthenticationEventHandler delegate (System.Web.Security)

ToString

*Description*

Represents the method that will handle the **FormsAuthentication** event of a **FormsAuthenticationModule**. The source of the event. A **System.Web.Security.FormsAuthenticationEventArgs** that contains the event data.

When you create a **System.Web.Security.FormsAuthenticationEventHandler** delegate, you identify the method that will handle the event. To associate the event with your event handler, add an instance of the delegate to the event. The event handler is called whenever the event occurs, unless you remove the delegate. For more information about event handler delegates, see This delegate defines the signature for the **FormsAuthentication\_OnAuthenticate** event handler.

FormsAuthenticationModule class (System.Web.Security)

ToString

*Description*

Enables ASP.NET applications to use forms authentication. This class cannot be inherited.

FormsAuthenticationModule

*Example Syntax:*

ToString

```
[C#] public FormsAuthenticationModule();  
[C++] public: FormsAuthenticationModule();  
[VB] Public Sub New()  
[JScript] public function FormsAuthenticationModule();
```

ToString

```
[C#] public event FormsAuthenticationEventHandler Authenticate;  
[C++] public: __event FormsAuthenticationEventHandler* Authenticate;  
[VB] Public Event Authenticate As FormsAuthenticationEventHandler
```

### *Description*

Defines the event raised during authentication. This is a Global.asax event that must be named **FormsAuthentication\_OnAuthenticate** . You can use this event to customize cookie authentication.

For more information about handling events, see .

Dispose

```
[C#] public void Dispose();  
[C++] public: __sealed void Dispose();  
[VB] NotOverridable Public Sub Dispose()  
[JScript] public function Dispose();
```

### *Description*

Disposes of the module derived from **IHttpModule** when called by the **HttpRuntime** .

Call **System.Web.Security.FormsAuthenticationModule.Dispose** when you are finished using the **System.Web.Security.FormsAuthenticationModule** .

The **System.Web.Security.FormsAuthenticationModule.Dispose** method leaves the **System.Web.Security.FormsAuthenticationModule** in an unusable state.

After calling **System.Web.Security.FormsAuthenticationModule.Dispose** , you must release all references to the

**System.Web.Security.FormsAuthenticationModule** so the memory it was occupying can be reclaimed by garbage collection.

#### Init

[C#] public void Init(HttpApplication app);

[C++] public: \_\_sealed void Init(HttpApplication\* app);

[VB] NotOverridable Public Sub Init(ByVal app As HttpApplication)

[JScript] public function Init(app : HttpApplication);

#### *Description*

Initializes the module derived from **IHttpModule** when called by the **HttpRuntime**. The **HttpApplication** module.

#### OnEnter

[C#] public void OnEnter(object source, EventArgs eventArgs);

[C++] public: void OnEnter(Object\* source, EventArgs\* eventArgs);

[VB] Public Sub OnEnter(ByVal source As Object, ByVal eventArgs As

EventArgs)

[JScript] public function OnEnter(source : Object, EventArgs : EventArgs);

*Description*

Called by the **HttpRuntime** . The source of the event. A **System.Web.Security.FormsAuthenticationEventArgs** that contains the event data.

OnLeave

[C#] public void OnLeave(object source, EventArgs EventArgs);

[C++] public: void OnLeave(Object\* source, EventArgs\* EventArgs);

[VB] Public Sub OnLeave(ByVal source As Object, ByVal EventArgs As EventArgs)

[JScript] public function OnLeave(source : Object, EventArgs : EventArgs);

*Description*

Called by the **HttpRuntime** . The source of the event. A **System.Web.Security.FormsAuthenticationEventArgs** that contains the event data.

FormsAuthenticationTicket class (System.Web.Security)

ToString

*Description*

Provides the information represented in an authentication cookie as used by **FormsAuthenticationModule** . This class cannot be inherited.

For more information about using attributes, see .

**FormsAuthenticationTicket**

*Example Syntax:*

**ToString**

[C#] public FormsAuthenticationTicket(string name, bool isPersistent, int timeout);

[C++] public: FormsAuthenticationTicket(String\* name, bool isPersistent, int timeout);

[VB] Public Sub New(ByVal name As String, ByVal isPersistent As Boolean, ByVal timeout As Integer)

[JScript] public function FormsAuthenticationTicket(name : String, isPersistent : Boolean, timeout : int);

### *Description*

Creates a **FormsAuthenticationTicket** instance with the specified name and cookie durability, and default values for the other settings (version set internally, dates set to current date, and expiration determined on durability of cookie).

The default settings include version number, dates that are set to the current date, and expiration as determined by the durability setting. User name associated with the ticket. If true, use a persistent cookie. Specifies the time in minutes (as an integer) for which the authentication ticket is valid.

FormsAuthenticationTicket

*Example Syntax:*

ToString

```
[C#] public FormsAuthenticationTicket(int version, string name, DateTime  
issueDate, DateTime expiration, bool isPersistent, string userData);
```

```
[C++] public: FormsAuthenticationTicket(int version, String* name, DateTime  
issueDate, DateTime expiration, bool isPersistent, String* userData);
```

```
[VB] Public Sub New(ByVal version As Integer, ByVal name As String, ByVal  
issueDate As DateTime, ByVal expiration As DateTime, ByVal isPersistent As  
Boolean, ByVal userData As String)
```

```
[JScript] public function FormsAuthenticationTicket(version : int, name : String,  
issueDate : DateTime, expiration : DateTime, isPersistent : Boolean, userData :  
String); Creates a FormsAuthenticationTicket instance.
```

### *Description*

Creates a **FormsAuthenticationTicket** instance with explicit values. The version number. User name associated with the ticket. Time at which the cookie was issued. Set an expiration date for the cookie. **True** if the cookie is persistent. User-defined data to be stored in the cookie.

FormsAuthenticationTicket

*Example Syntax:*

ToString

```
[C#] public FormsAuthenticationTicket(int version, string name, DateTime
```

```

1  issueDate, DateTime expiration, bool isPersistent, string userData, string
2  cookiePath);
3  [C++] public: FormsAuthenticationTicket(int version, String* name, DateTime
4  issueDate, DateTime expiration, bool isPersistent, String* userData, String*
5  cookiePath);
6  [VB] Public Sub New(ByVal version As Integer, ByVal name As String, ByVal
7  issueDate As DateTime, ByVal expiration As DateTime, ByVal isPersistent As
8  Boolean, ByVal userData As String, ByVal cookiePath As String)
9  [JScript] public function FormsAuthenticationTicket(version : int, name : String,
10 issueDate : DateTime, expiration : DateTime, isPersistent : Boolean, userData :
11 String, cookiePath : String);
12

```

### *Description*

Creates a **FormsAuthenticationTicket** instance with explicit values. The version number. User name associated with the ticket. Time at which the cookie was issued. Set an expiration date for the cookie. **True** if the cookie is persistent. User-defined data to be stored in the cookie. The path for the cookie.

CookiePath

ToString

```

21 [C#] public string CookiePath {get;}
22 [C++] public: __property String* get_CookiePath();
23 [VB] Public ReadOnly Property CookiePath As String
24 [JScript] public function get CookiePath() : String;
25

```

1  
2 *Description*

3 Returns the Path for which the cookie was issued.

4 This is used when the cookie is refreshed.

5 Expiration

6 ToString

7  
8 [C#] public DateTime Expiration {get;}

9 [C++] public: \_\_property DateTime get\_Expiration();

10 [VB] Public ReadOnly Property Expiration As DateTime

11 [JScript] public function get Expiration() : DateTime;

12  
13 *Description*

14 Returns the date/time at which the cookie expires.

15 For durable cookies, this should be the maximum value. For session scoped  
16 cookies, this should be set to the time the cookie was issued. This can be used by  
17 custom application logic to implement more-advanced expiration semantics.

18 Expired

19 ToString

20  
21 [C#] public bool Expired {get;}

22 [C++] public: \_\_property bool get\_Expired();

23 [VB] Public ReadOnly Property Expired As Boolean

24 [JScript] public function get Expired() : Boolean;



1  
2 *Description*

3 Returns **True** if the cookie has expired.

4 **IsPersistent**

5 **ToString**

6  
7 [C#] public bool IsPersistent {get;}

8 [C++] public: \_\_property bool get\_IsPersistent();

9 [VB] Public ReadOnly Property IsPersistent As Boolean

10 [JScript] public function get IsPersistent() : Boolean;

11  
12 *Description*

13 Returns **True** if a durable cookie was issued. Otherwise, the authentication  
14 cookie is scoped to the browser lifetime.

15 **IssueDate**

16 **ToString**

17  
18 [C#] public DateTime IssueDate {get;}

19 [C++] public: \_\_property DateTime get\_IssueDate();

20 [VB] Public ReadOnly Property IssueDate As DateTime

21 [JScript] public function get IssueDate() : DateTime;

22  
23 *Description*

24 Returns the date/time at which the cookie was originally issued. This can be  
25 used for custom expiration schemes.

Name

ToString

[C#] public string Name {get;}

[C++] public: \_\_property String\* get\_Name();

[VB] Public ReadOnly Property Name As String

[JScript] public function get Name() : String;

*Description*

Returns the user name associated with the authentication cookie.

This string is used for authorization purposes.

UserData

ToString

[C#] public string UserData {get;}

[C++] public: \_\_property String\* get\_UserData();

[VB] Public ReadOnly Property UserData As String

[JScript] public function get UserData() : String;

*Description*

Returns an application-defined string that might have been stored in the cookie.

This field will be empty ("" ) if no application-defined data was provided.

Version

ToString

```

1
2 [C#] public int Version {get;}
3 [C++] public: __property int get_Version();
4 [VB] Public ReadOnly Property Version As Integer
5 [JScript] public function get Version() : int;
6

```

### *Description*

Returns a byte version number for future use.

The current version identifier is 1.

FormsIdentity class (System.Web.Security)

ToString

### *Description*

Provides an **IIdentity**- derived class to be used by

**FormsAuthenticationModule** . It provides a way for an application to access the cookie authentication ticket. This class cannot be inherited.

FormsIdentity

*Example Syntax:*

ToString

```

22 [C#] public FormsIdentity(FormsAuthenticationTicket ticket);
23 [C++] public: FormsIdentity(FormsAuthenticationTicket* ticket);
24 [VB] Public Sub New(ByVal ticket As FormsAuthenticationTicket)
25 [JScript] public function FormsIdentity(ticket : FormsAuthenticationTicket);

```

1  
2 *Description*

3        Initializes a new instance of the **System.Web.Security.FormsIdentity**  
4 class. The authentication ticket upon which this identity is based.

5        AuthenticationType

6        ToString

7  
8 [C#] public string AuthenticationType {get;}

9 [C++] public: \_\_property String\* get\_AuthenticationType();

10 [VB] Public ReadOnly Property AuthenticationType As String

11 [JScript] public function get AuthenticationType() : String;

12  
13 *Description*

14        The type of the identity (in this case, "Forms").

15        IsAuthenticated

16        ToString

17  
18 [C#] public bool IsAuthenticated {get;}

19 [C++] public: \_\_property bool get\_IsAuthenticated();

20 [VB] Public ReadOnly Property IsAuthenticated As Boolean

21 [JScript] public function get IsAuthenticated() : Boolean;

22  
23 *Description*

24        Indicates whether authentication took place.

25        Name

ToString

[C#] public string Name {get;}

[C++] public: \_\_property String\* get\_Name();

[VB] Public ReadOnly Property Name As String

[JScript] public function get Name() : String;

*Description*

The name of the identity (in this case, the user name).

Ticket

ToString

[C#] public FormsAuthenticationTicket Ticket {get;}

[C++] public: \_\_property FormsAuthenticationTicket\* get\_Ticket();

[VB] Public ReadOnly Property Ticket As FormsAuthenticationTicket

[JScript] public function get Ticket() : FormsAuthenticationTicket;

*Description*

Returns the **FormsAuthenticationTicket** associated with the current request.

PassportAuthenticationEventArgs class (System.Web.Security)

ToString

*Description*

1 The event argument passed to the  
2 **PassportAuthentication\_OnAuthenticate** event by the **PassportAuthentication**  
3 module. Since there is already an identity at this point, this is useful mainly for  
4 attaching a custom **IPrincipal** object to the context using the supplied identity.

5 For more information about handling events, see .

6 **PassportAuthenticationEventArgs**

7 *Example Syntax:*

8 **ToString**

9  
10 [C#] public PassportAuthenticationEventArgs(PassportIdentity identity,  
11 HttpContext context);

12 [C++] public: PassportAuthenticationEventArgs(PassportIdentity\* identity,  
13 HttpContext\* context);

14 [VB] Public Sub New(ByVal identity As PassportIdentity, ByVal context As  
15 HttpContext)

16 [JScript] public function PassportAuthenticationEventArgs(identity :  
17 PassportIdentity, context : HttpContext);

18  
19 *Description*

20 Initializes a new instance of the  
21 **System.Web.Security.PassportAuthenticationEventArgs** class. The identity  
22 object The context for the event.

23 **Context**

24 **ToString**

```

1
2 [C#] public HttpContext Context {get;}
3 [C++] public: __property HttpContext* get_Context();
4 [VB] Public ReadOnly Property Context As HttpContext
5 [JScript] public function get Context() : HttpContext;
6

```

### *Description*

The **HttpContext** intrinsic provides access to **Request** , **Response** , and **User** objects.

Identity

ToString

```

13 [C#] public PassportIdentity Identity {get;}
14 [C++] public: __property PassportIdentity* get_Identity();
15 [VB] Public ReadOnly Property Identity As PassportIdentity
16 [JScript] public function get Identity() : PassportIdentity;
17

```

### *Description*

An authenticated Passport identity.

User

ToString

```

23 [C#] public IPrincipal User {get; set;}
24 [C++] public: __property IPrincipal* get_User();public: __property void
25 set_User(IPrincipal*);

```

1 [VB] Public Property User As IPrincipal

2 [JScript] public function get User() : IPrincipal;public function set

3 User(IPrincipal);

4  
5 *Description*

6 Associates an **IPrincipal** object with the request.

7 The **User** object should be attached to the context. If User is non-null and  
8 **Context.User** is null, the **PassportAuthenticationModule** will initialize  
9 **Context.User** with **PassportAuthenticationEventArgs.User** .

10 PassportAuthenticationEventHandler delegate (System.Web.Security)

11 ToString

12  
13  
14 *Description*

15 Represents the method that handles the  
16 **PassportAuthentication\_OnAuthenticate** event of a  
17 **PassportAuthenticationModule** . The object that raised the event. A  
18 **PassportAuthenticationEventArgs** object that contains the event data.

19 When you create a  
20 **System.Web.Security.PassportAuthenticationEventHandler** delegate, you  
21 identify the method to handle the event. To associate the event with your  
22 EventHandler, add an instance of the delegate to the event. The EventHandler is  
23 called whenever the event occurs, unless you remove the delegate. For more  
24 information about EventHandler delegates, see .

25 PassportAuthenticationModule class (System.Web.Security)



ToString

*Description*

Provides a wrapper around PassportAuthentication services. This class cannot be inherited.

Installation of the Passport SDK is still required, as is registration of the site with the Passport authority. The **PassportAuthentication\_OnAuthenticate** event is raised for applications that are designed to attach a custom **IPrincipal** object to the context. The Passport service itself does the authentication, so that cannot be overridden.

PassportAuthenticationModule

*Example Syntax:*

ToString

```
[C#] public PassportAuthenticationModule();
```

```
[C++] public: PassportAuthenticationModule();
```

```
[VB] Public Sub New()
```

```
[JScript] public function PassportAuthenticationModule();
```

ToString

```
[C#] public event PassportAuthenticationEventHandler Authenticate;
```

```
[C++] public: __event PassportAuthenticationEventHandler* Authenticate;
```

```
[VB] Public Event Authenticate As PassportAuthenticationEventHandler
```

## *Description*

Raised during authentication. This is a Global.asax event that must be named **PassportAuthentication\_OnAuthenticate** event.

Advanced users use this event to attach a custom **IPrinciple** object to the context.

## **Dispose**

[C#] public void Dispose();

[C++] public: \_\_sealed void Dispose();

[VB] NotOverridable Public Sub Dispose()

[JScript] public function Dispose();

## *Description*

Disposes of the module derived from **IHttpModule** when called by the **HttpRuntime**.

Call **System.Web.Security.PassportAuthenticationModule.Dispose** when you are finished using the **System.Web.Security.PassportAuthenticationModule**. The **System.Web.Security.PassportAuthenticationModule.Dispose** method leaves the **System.Web.Security.PassportAuthenticationModule** in an unusable state. After calling **System.Web.Security.PassportAuthenticationModule.Dispose**, you must release all references to the **System.Web.Security.PassportAuthenticationModule** so the memory it occupied can be reclaimed by garbage collection.

## Init

```
[C#] public void Init(HttpApplication app);  
[C++] public: __sealed void Init(HttpApplication* app);  
[VB] NotOverridable Public Sub Init(ByVal app As HttpApplication)  
[JScript] public function Init(app : HttpApplication);
```

### *Description*

Initializes the module (derived from **IHttpModule** ) when called by the **HttpRuntime** . The **HttpApplication** module

PassportIdentity class (System.Web.Security)

ToString

### *Description*

Provides access to the Passport profile information contained in the Passport profile cookies. This is an **IIdentity**- derived class. This class cannot be inherited.

ASP.NET Beta 2 requires the 1.4 version of the Passport SDK.

PassportIdentity

*Example Syntax:*

ToString

```
[C#] public PassportIdentity();  
[C++] public: PassportIdentity();
```

1 [VB] Public Sub New()

2 [JScript] public function PassportIdentity();

3  
4 *Description*

5 Creates a new instance of the **PassportIdentity** object.

6 AuthenticationType

7 ToString

8  
9 [C#] public string AuthenticationType {get;}

10 [C++] public: \_\_property String\* get\_AuthenticationType();

11 [VB] Public ReadOnly Property AuthenticationType As String

12 [JScript] public function get AuthenticationType() : String;

13  
14 *Description*

15 The type of the identity. In this case, **Passport** .

16 See the Passport SDK documentation for more information about

17 **PassportIdentity.AuthenticationType** .

18 Error

19 ToString

20  
21 [C#] public int Error {get;}

22 [C++] public: \_\_property int get\_Error();

23 [VB] Public ReadOnly Property Error As Integer

24 [JScript] public function get Error() : int;

1  
2 *Description*

3 Returns an error state associated with the current Passport ticket. For more  
4 information, see the error property in the Passport documentation .

5 GetFromNetworkServer

6 ToString

7  
8 [C#] public bool GetFromNetworkServer {get;}

9 [C++] public: \_\_property bool get\_GetFromNetworkServer();

10 [VB] Public ReadOnly Property GetFromNetworkServer As Boolean

11 [JScript] public function get GetFromNetworkServer() : Boolean;

12  
13 *Description*

14 Returns **true** if a connection is coming back from the Passport server  
15 (logon, update, or registration) and if the Passport data contained on the query  
16 string is valid.

17 HasSavedPassword

18 ToString

19  
20 [C#] public bool HasSavedPassword {get;}

21 [C++] public: \_\_property bool get\_HasSavedPassword();

22 [VB] Public ReadOnly Property HasSavedPassword As Boolean

23 [JScript] public function get HasSavedPassword() : Boolean;

24  
25 *Description*

1 Returns **true** if the Passport member's ticket indicates that the password  
2 was saved on the Passport logon page the last time the ticket was refreshed.

3 For more information, see the Passport SDK documentation .

4 HasTicket

5 ToString

6  
7 [C#] public bool HasTicket {get;}

8 [C++] public: \_\_property bool get\_HasTicket();

9 [VB] Public ReadOnly Property HasTicket As Boolean

10 [JScript] public function get HasTicket() : Boolean;

11  
12 *Description*

13 Returns **true** if there is a Passport ticket as a cookie on the query string.

14 For more information, see the Passport SDK documentation .

15 IsAuthenticated

16 ToString

17  
18 [C#] public bool IsAuthenticated {get;}

19 [C++] public: \_\_property bool get\_IsAuthenticated();

20 [VB] Public ReadOnly Property IsAuthenticated As Boolean

21 [JScript] public function get IsAuthenticated() : Boolean;

22  
23 *Description*

24 Returns **true** if the user is authenticated by a Passport authority.

25 For more information, see the Passport SDK documentation .

Item

ToString

[C#] public string this[string strProfileName] {get;}

[C++] public: \_\_property String\* get\_Item(String\* strProfileName);

[VB] Public Default ReadOnly Property Item(ByVal strProfileName As String)

As String

[JScript] returnValue = PassportIdentityObject.Item(strProfileName);

### *Description*

Accesses Passport profile attributes . Calling this property is the equivalent of calling **GetProfileObject** or **SetProfileObject** .

For more information, see the Passport SDK documentation . The Passport profile attribute to query.

Name

ToString

[C#] public string Name {get;}

[C++] public: \_\_property String\* get\_Name();

[VB] Public ReadOnly Property Name As String

[JScript] public function get Name() : String;

### *Description*

Consists of the name of the identity. In this case, it is the value of the Passport PUID.

For more information, see the Passport SDK documentation .

TicketAge

ToString

[C#] public int TicketAge {get;}

[C++] public: \_\_property int get\_TicketAge();

[VB] Public ReadOnly Property TicketAge As Integer

[JScript] public function get TicketAge() : int;

#### *Description*

Consists of the time, in seconds, since the last ticket was issued or refreshed.

For more information, see the Passport SDK documentation .

TimeSinceSignIn

ToString

[C#] public int TimeSinceSignIn {get;}

[C++] public: \_\_property int get\_TimeSinceSignIn();

[VB] Public ReadOnly Property TimeSinceSignIn As Integer

[JScript] public function get TimeSinceSignIn() : int;

#### *Description*

Consists of the time, in seconds, since a member's log on to the Passport logon server.

For more information, see the Passport SDK documentation .



## AuthUrl

[C#] public string AuthUrl();

[C++] public: String\* AuthUrl();

[VB] Public Function AuthUrl() As String

[JScript] public function AuthUrl() : String; There are two overloads for this method.

### *Description*

Returns a string containing the login server URL for a member, along with the optional information sent to the login server in the query string.

## AuthUrl

[C#] public string AuthUrl(string strReturnUrl, int iTimeWindow, bool fForceLogin, string strCoBrandedArgs, int iLangID, string strNameSpace, int iKPP, bool bUseSecureAuth);

[C++] public: String\* AuthUrl(String\* strReturnUrl, int iTimeWindow, bool fForceLogin, String\* strCoBrandedArgs, int iLangID, String\* strNameSpace, int iKPP, bool bUseSecureAuth);

[VB] Public Function AuthUrl(ByVal strReturnUrl As String, ByVal iTimeWindow As Integer, ByVal fForceLogin As Boolean, ByVal strCoBrandedArgs As String, ByVal iLangID As Integer, ByVal strNameSpace As String, ByVal iKPP As Integer, ByVal bUseSecureAuth As Boolean) As String

[JScript] public function AuthUrl(strReturnUrl : String, iTimeWindow : int, fForceLogin : Boolean, strCoBrandedArgs : String, iLangID : int, strNameSpace :

String, iKPP : int, bUseSecureAuth : Boolean) : String;

### *Description*

Similar to AuthUrl(). Returns the authentication server URL for a member.

For more information about the parameters for this method, see the Passport SDK documentation. Sets the URL of the location that the Login server should redirect to after sign-in is complete. Specifies the interval during which members must have last signed in. Determines how the iTimeWindow will be used. Specifies variables to be appended to the URL of the Co-branding Template script page that was specified at initial participant registration. Specifies the language in which the required domain authority page should be displayed. Specifies the domain in which the Passport should be created. Specifies data collection policies for purposes of COPPA compliance. Declares whether the actual sign-in UI should be served HTTPS from the Passport domain authority.

### *AuthUrl*

[C#] public string AuthUrl(string strReturnUrl, int iTimeWindow, int

iForceLogin, string strCoBrandedArgs, int iLangID, string strNameSpace, int

iKPP, int iUseSecureAuth);

[C++] public: String\* AuthUrl(String\* strReturnUrl, int iTimeWindow, int

iForceLogin, String\* strCoBrandedArgs, int iLangID, String\* strNameSpace, int

iKPP, int iUseSecureAuth);

[VB] Public Function AuthUrl(ByVal strReturnUrl As String, ByVal

iTimeWindow As Integer, ByVal iForceLogin As Integer, ByVal

strCoBrandedArgs As String, ByVal iLangID As Integer, ByVal strNameSpace

As String, ByVal iKPP As Integer, ByVal iUseSecureAuth As Integer) As String  
[JScript] public function AuthUrl(strReturnUrl : String, iTimeWindow : int,  
iForceLogin : int, strCoBrandedArgs : String, iLangID : int, strNameSpace :  
String, iKPP : int, iUseSecureAuth : int) : String;

### *Description*

#### AuthUrl2

[C#] public string AuthUrl2();  
[C++] public: String\* AuthUrl2();  
[VB] Public Function AuthUrl2() As String  
[JScript] public function AuthUrl2() : String; Retrieves a **String** containing the  
logon server URL for a member, as well as the optional information sent to the  
logon server in the query string.

### *Description*

Retrieves a **String** containing the logon server URL for a member, as well  
as the optional information sent to the logon server in the query string.

*Return Value:* The logon server URL for a member, as well as the optional  
information sent to the logon server in the query string.

This URL can be used to generate a link for a member who has not signed  
in previously or who has an expired ticket. For more details, see  
**IPassportManager2::AuthUrl2** in the Passport SDK documentation.

#### AuthUrl2

```

1
2 [C#] public string AuthUrl2(string strReturnUrl, int iTimeWindow, bool
3 fForceLogin, string strCoBrandedArgs, int iLangID, string strNameSpace, int
4 iKPP, bool bUseSecureAuth);
5 [C++] public: String* AuthUrl2(String* strReturnUrl, int iTimeWindow, bool
6 fForceLogin, String* strCoBrandedArgs, int iLangID, String* strNameSpace, int
7 iKPP, bool bUseSecureAuth);
8 [VB] Public Function AuthUrl2(ByVal strReturnUrl As String, ByVal
9 iTimeWindow As Integer, ByVal fForceLogin As Boolean, ByVal
10 strCoBrandedArgs As String, ByVal iLangID As Integer, ByVal strNameSpace
11 As String, ByVal iKPP As Integer, ByVal bUseSecureAuth As Boolean) As String
12 [JScript] public function AuthUrl2(strReturnUrl : String, iTimeWindow : int,
13 fForceLogin : Boolean, strCoBrandedArgs : String, iLangID : int, strNameSpace :
14 String, iKPP : int, bUseSecureAuth : Boolean) : String;
15

```

### *Description*

Retrieves a **String** containing the logon server URL for a member, as well as the optional information sent to the logon server in the query string.

*Return Value:* The logon server URL for a member, as well as the optional information sent to the logon server in the query string.

This URL can be used to generate a link for a member who has not signed in previously or who has an expired ticket. For more details, see **IPassportManager2::AuthUrl2** in the Passport SDK documentation. See Passport documentation for **IPassportManager2::AuthUrl2**. See Passport documentation for **IPassportManager2::AuthUrl2**. See Passport documentation

1 for **IPassportManager2::AuthUrl2**. See Passport documentation for  
2 **IPassportManager2::AuthUrl2**. See Passport documentation for  
3 **IPassportManager2::AuthUrl2**. See Passport documentation for  
4 **IPassportManager2::AuthUrl2**. See Passport documentation for  
5 **IPassportManager2::AuthUrl2**. See Passport documentation for  
6 **IPassportManager2::AuthUrl2**.

7       AuthUrl2

8  
9 [C#] public string AuthUrl2(string strReturnUrl, int iTimeWindow, int  
10 iForceLogin, string strCoBrandedArgs, int iLangID, string strNameSpace, int  
11 iKPP, int iUseSecureAuth);  
12 [C++] public: String\* AuthUrl2(String\* strReturnUrl, int iTimeWindow, int  
13 iForceLogin, String\* strCoBrandedArgs, int iLangID, String\* strNameSpace, int  
14 iKPP, int iUseSecureAuth);  
15 [VB] Public Function AuthUrl2(ByVal strReturnUrl As String, ByVal  
16 iTimeWindow As Integer, ByVal iForceLogin As Integer, ByVal  
17 strCoBrandedArgs As String, ByVal iLangID As Integer, ByVal strNameSpace  
18 As String, ByVal iKPP As Integer, ByVal iUseSecureAuth As Integer) As String  
19 [JScript] public function AuthUrl2(strReturnUrl : String, iTimeWindow : int,  
20 iForceLogin : int, strCoBrandedArgs : String, iLangID : int, strNameSpace :  
21 String, iKPP : int, iUseSecureAuth : int) : String;

22  
23 *Description*

24       Retrieves a **String** containing the logon server URL for a member, as well  
25 as the optional information sent to the logon server in the query string.

1 *Return Value:* The logon server URL for a member, as well as the optional  
2 information sent to the logon server in the query string.

3       You can use this URL to generate a link for a member who has not  
4 previously signed in or who has an expired ticket. For more details, see  
5 **IPassportManager2::AuthUrl2** in the Passport SDK documentation. See  
6 Passport documentation for **IPassportManager2::AuthUrl2**. See Passport  
7 documentation for **IPassportManager2::AuthUrl2**. See Passport documentation  
8 for **IPassportManager2::AuthUrl2**. See Passport documentation for  
9 **IPassportManager2::AuthUrl2**. See Passport documentation for  
10 **IPassportManager2::AuthUrl2**. See Passport documentation for  
11 **IPassportManager2::AuthUrl2**. See Passport documentation for  
12 **IPassportManager2::AuthUrl2**.

### 13       Compress

14  
15 [C#] public static string Compress(string strData);  
16 [C++] public: static String\* Compress(String\* strData);  
17 [VB] Public Shared Function Compress(ByVal strData As String) As String  
18 [JScript] public static function Compress(strData : String) : String;  
19

### 20 *Description*

21       Compresses data.

22 *Return Value:* Compressed data.

23       For more information about **IPassportCrypt::Compress**, see the Passport  
24 SDK documentation . Data to be compressed.

### 25       CryptIsValid

1  
2 [C#] public static bool CryptIsValid();

3 [C++] public: static bool CryptIsValid();

4 [VB] Public Shared Function CryptIsValid() As Boolean

5 [JScript] public static function CryptIsValid() : Boolean;

6  
7 *Description*

8 Returns **true** if the key used for encryption and decryption is valid and if  
9 the Passport **Manager** object is in a valid state for encryption.

10 See the Passport SDK documentation for **IPassportCrypt::get\_IsValid** .

11 CryptPutHost

12  
13 [C#] public static int CryptPutHost(string strHost);

14 [C++] public: static int CryptPutHost(String\* strHost);

15 [VB] Public Shared Function CryptPutHost(ByVal strHost As String) As Integer

16 [JScript] public static function CryptPutHost(strHost : String) : int;

17  
18 *Description*

19 Sets the key being used by the current Passport **Crypt** object by referring to  
20 the host name or IP address used by the desired installation.

21 See the Passport SDK documentation for **IPassportCrypt::put\_Host** .

22 Host name or IP address.

23 CryptPutSite

24  
25 [C#] public static int CryptPutSite(string strSite);

1 [C++] public: static int CryptPutSite(String\* strSite);

2 [VB] Public Shared Function CryptPutSite(ByVal strSite As String) As Integer

3 [JScript] public static function CryptPutSite(strSite : String) : int;

4  
5 *Description*

6 Sets the key being used by the current Passport **Crypt** object by referring to  
7 the site-name label assigned to that key when the key was first installed.

8 See the Passport SDK documentation for **IPassportCrypt::put\_Site** .

9 **Decompress**

10  
11 [C#] public static string Decompress(string strData);

12 [C++] public: static String\* Decompress(String\* strData);

13 [VB] Public Shared Function Decompress(ByVal strData As String) As String

14 [JScript] public static function Decompress(strData : String) : String;

15  
16 *Description*

17 Decompresses data that has been compressed by the **Compress** method..

18 *Return Value:* Decompressed data.

19 See the Passport SDK documentation for **IPassportCrypt::Compress** .

20 Data to be decompressed.

21 **Decrypt**

22  
23 [C#] public static string Decrypt(string strData);

24 [C++] public: static String\* Decrypt(String\* strData);

25 [VB] Public Shared Function Decrypt(ByVal strData As String) As String



1 [JScript] public static function Decrypt(strData : String) : String;

2  
3 *Description*

4       Decrypts data using the Passport participant key for the current site.

5 *Return Value:* Decrypted data.

6       See the Passport SDK documentation for **IPassportCrypt::Decrypt** . The  
7 data to be decrypted.

8       Encrypt

9  
10 [C#] public static string Encrypt(string strData);

11 [C++] public: static String\* Encrypt(String\* strData);

12 [VB] Public Shared Function Encrypt(ByVal strData As String) As String

13 [JScript] public static function Encrypt(strData : String) : String;

14  
15 *Description*

16       Encrypts data using the Passport participant key for the current site.

17       Maximum input size is 2045 characters.

18 *Return Value:* Encrypted data.

19       See the Passport SDK documentation for **IPassportCrypt::Encrypt** . The  
20 data to be encrypted.

21       Finalize

22  
23 [C#] ~PassportIdentity();

24 [C++] ~PassportIdentity();

25 [VB] Overrides Protected Sub Finalize()

1 [JScript] protected override function Finalize();

2  
3 *Description*

4       Calls **Finalize** method.

5       GetDomainAttribute

6  
7 [C#] public string GetDomainAttribute(string strAttribute, int iLCID, string  
8 strDomain);

9 [C++] public: String\* GetDomainAttribute(String\* strAttribute, int iLCID, String\*  
10 strDomain);

11 [VB] Public Function GetDomainAttribute(ByVal strAttribute As String, ByVal  
12 iLCID As Integer, ByVal strDomain As String) As String

13 [JScript] public function GetDomainAttribute(strAttribute : String, iLCID : int,  
14 strDomain : String) : String;

15  
16 *Description*

17       Provides information for a Passport domain by querying the Passport CCD  
18 for the requested **DomainAttribute** .

19       For more information, see the Passport SDK documentation . The name of  
20 the attribute value to retrieve. Specifies the language in which various Passport  
21 network pages should be displayed to the member. The domain authority name to  
22 query for an attribute.

23       GetDomainFromMemberName

24  
25 [C#] public string GetDomainFromMemberName(string strMemberName);

1 [C++] public: String\* GetDomainFromMemberName(String\* strMemberName);  
2 [VB] Public Function GetDomainFromMemberName(ByVal strMemberName As  
3 String) As String  
4 [JScript] public function GetDomainFromMemberName(strMemberName :  
5 String) : String;

6

7 *Description*

8 Returns the Passport domain from the member-name string. Name of the  
9 Passport member

10 GetIsAuthenticated

11

12 [C#] public bool GetIsAuthenticated(int iTimeWindow, bool bForceLogin, bool  
13 bCheckSecure);

14 [C++] public: bool GetIsAuthenticated(int iTimeWindow, bool bForceLogin, bool  
15 bCheckSecure);

16 [VB] Public Function GetIsAuthenticated(ByVal iTimeWindow As Integer,  
17 ByVal bForceLogin As Boolean, ByVal bCheckSecure As Boolean) As Boolean

18 [JScript] public function GetIsAuthenticated(iTimeWindow : int, bForceLogin :  
19 Boolean, bCheckSecure : Boolean) : Boolean; Returns **true** if the user is  
20 authenticated by a Passport authority.

21

22 *Description*

23 Returns **true** if the user is authenticated by a Passport authority.

24 *Return Value:* **true** if the user is authenticated by a Passport authority.

25

1 See passport documentation for **IPassportManager::IsAuthenticated** .  
2 Specifies the interval during which members must have last signed in to the  
3 calling domain. Determines how *iTimeWindow* is used. Enables checking for a  
4 secure logon.

#### 5 GetIsAuthenticated

6  
7 [C#] public bool GetIsAuthenticated(int iTimeWindow, int iForceLogin, int  
8 iCheckSecure);  
9 [C++] public: bool GetIsAuthenticated(int iTimeWindow, int iForceLogin, int  
10 iCheckSecure);  
11 [VB] Public Function GetIsAuthenticated(ByVal iTimeWindow As Integer,  
12 ByVal iForceLogin As Integer, ByVal iCheckSecure As Integer) As Boolean  
13 [JScript] public function GetIsAuthenticated(iTimeWindow : int, iForceLogin :  
14 int, iCheckSecure : int) : Boolean;

#### 16 Description

17 Returns **true** if the user is authenticated by a Passport authority.

18 *Return Value:* **true** if the user is authenticated by a Passport authority.

19 See Passport documentation for **IPassportManager::IsAuthenticated** .  
20 Specifies the interval during which members must have last signed in to the  
21 calling domain. Determines how *iTimeWindow* is used. Enables checking for a  
22 secure logon.

#### 23 GetProfileObject

24  
25 [C#] public object GetProfileObject(string strProfileName);

1 [C++] public: Object\* GetProfileObject(String\* strProfileName);

2 [VB] Public Function GetProfileObject(ByVal strProfileName As String) As

3 Object

4 [JScript] public function GetProfileObject(strProfileName : String) : Object;

5  
6 *Description*

7 Returns Passport profile information for the supplied profile attribute. The  
8 Passport profile attribute to query.

9 HasFlag

10  
11 [C#] public bool HasFlag(int iFlagMask);

12 [C++] public: bool HasFlag(int iFlagMask);

13 [VB] Public Function HasFlag(ByVal iFlagMask As Integer) As Boolean

14 [JScript] public function HasFlag(iFlagMask : int) : Boolean;

15  
16 *Description*

17 Returns **true** if a given flag is set in this user's profile.

18 For more information, see the Passport SDK documentation . The Passport  
19 profile flag to query.

20 HasProfile

21  
22 [C#] public bool HasProfile(string strProfile);

23 [C++] public: bool HasProfile(String\* strProfile);

24 [VB] Public Function HasProfile(ByVal strProfile As String) As Boolean

25 [JScript] public function HasProfile(strProfile : String) : Boolean;

## *Description*

Returns **true** if a given profile attribute exists in this user's profile.

For more information, see the Passport SDK documentation . The Passport profile attribute to query.

### **HaveConsent**

```
[C#] public bool HaveConsent(bool bNeedFullConsent, bool bNeedBirthdate);
```

```
[C++] public: bool HaveConsent(bool bNeedFullConsent, bool bNeedBirthdate);
```

```
[VB] Public Function HaveConsent(ByVal bNeedFullConsent As Boolean, ByVal  
bNeedBirthdate As Boolean) As Boolean
```

```
[JScript] public function HaveConsent(bNeedFullConsent : Boolean,  
bNeedBirthdate : Boolean) : Boolean;
```

## *Description*

Returns **true** if Full Consent is granted in this user's profile.

For more information, see the Passport SDK documentation . true if Full Consent is required for Passport Authentication.

### **LoginUser**

```
[C#] public int LoginUser();
```

```
[C++] public: int LoginUser();
```

```
[VB] Public Function LoginUser() As Integer
```

```
[JScript] public function LoginUser() : int;
```

## Description

Logs the user on, either by generating a 302-redirect URL or initiating a Passport-aware client authentication exchange.

This method supports Passport-aware client applications. Note that there are some policy restrictions on the use of this method in certain scenarios. Note also that this method should not be called from a frameset, as suggested in the Passport documentation. Finally, this method requires that Msppfltr.dll be properly installed and running as an ISAPI filter at the global level of the participant Web site.

### LoginUser

```
[C#] public int LoginUser(string szRetURL, int iTimeWindow, bool fForceLogin,
string szCOBrandArgs, int iLangID, string strNameSpace, int iKPP, bool
fUseSecureAuth, object oExtraParams);
```

```
[C++] public: int LoginUser(String* szRetURL, int iTimeWindow, bool
fForceLogin, String* szCOBrandArgs, int iLangID, String* strNameSpace, int
iKPP, bool fUseSecureAuth, Object* oExtraParams);
```

```
[VB] Public Function LoginUser(ByVal szRetURL As String, ByVal
iTimeWindow As Integer, ByVal fForceLogin As Boolean, ByVal
szCOBrandArgs As String, ByVal iLangID As Integer, ByVal strNameSpace As
String, ByVal iKPP As Integer, ByVal fUseSecureAuth As Boolean, ByVal
oExtraParams As Object) As Integer
```

```
[JScript] public function LoginUser(szRetURL : String, iTimeWindow : int,
fForceLogin : Boolean, szCOBrandArgs : String, iLangID : int, strNameSpace :
```

String, iKPP : int, fUseSecureAuth : Boolean, oExtraParams : Object) : int; Logs the user on, either by generating a 302-redirect URL or initiating a Passport-aware client authentication exchange.

### *Description*

Logs the user on, either by generating a 302-redirect URL or initiating a Passport-aware client authentication exchange.

This method supports Passport-aware client applications. Note that there are some policy restrictions on the use of this method in certain scenarios. Note also that this method should not be called from a frameset, as suggested in the Passport documentation. Finally, this method requires that Msppfltr.dll be properly installed and running as an ISAPI filter at the global level of the participant Web site. See Passport documentation for **IPassportManager2::LoginUser** . See Passport documentation for **IPassportManager2::LoginUser** . See Passport documentation for **IPassportManager2::LoginUser** . See Passport documentation for **IPassportManager2::LoginUser** . See Passport documentation for **IPassportManager2::LoginUser** . See Passport documentation for **IPassportManager2::LoginUser** . See Passport documentation for **IPassportManager2::LoginUser** .

### **LoginUser**

[C#] public int LoginUser(string szRetURL, int iTimeWindow, int fForceLogin, string szCOBrandArgs, int iLangID, string strNameSpace, int iKPP, int iUseSecureAuth, object oExtraParams);

[C++] public: int LoginUser(String\* szRetURL, int iTimeWindow, int



```

1 fForceLogin, String* szCOBrandArgs, int iLangID, String* strNameSpace, int
2 iKPP, int iUseSecureAuth, Object* oExtraParams);
3 [VB] Public Function LoginUser(ByVal szRetURL As String, ByVal
4 iTimeWindow As Integer, ByVal fForceLogin As Integer, ByVal szCOBrandArgs
5 As String, ByVal iLangID As Integer, ByVal strNameSpace As String, ByVal
6 iKPP As Integer, ByVal iUseSecureAuth As Integer, ByVal oExtraParams As
7 Object) As Integer
8 [JScript] public function LoginUser(szRetURL : String, iTimeWindow : int,
9 fForceLogin : int, szCOBrandArgs : String, iLangID : int, strNameSpace : String,
10 iKPP : int, iUseSecureAuth : int, oExtraParams : Object) : int;

```

### *Description*

Logs the user on, either by generating a 302-redirect URL or initiating a Passport-aware client authentication exchange.

This method supports Passport-aware client applications. Note that there are some policy restrictions on the use of this method in certain scenarios. Note also that this method should not be called from a frameset, as suggested in the Passport documentation. Finally, this method requires that Msppfltr.dll be properly installed and running as an ISAPI filter at the global level of the participant Web site. See Passport documentation for **IPassportManager2::LoginUser** . See Passport documentation for **IPassportManager2::LoginUser** . See Passport documentation for **IPassportManager2::LoginUser** . See Passport documentation for **IPassportManager2::LoginUser** . See Passport documentation for **IPassportManager2::LoginUser** . See Passport

1 documentation for **IPassportManager2::LoginUser** . See Passport  
2 documentation for **IPassportManager2::LoginUser** .

### 3 LogoTag

4  
5 [C#] public string LogoTag();  
6 [C++] public: String\* LogoTag();  
7 [VB] Public Function LogoTag() As String  
8 [JScript] public function LogoTag() : String;

### 9 10 *Description*

11 Returns an HTML snippet containing an image tag for a Passport link. This  
12 is based on the current state of the identity (already signed in, and such).

### 13 LogoTag

14  
15 [C#] public string LogoTag(string strReturnUrl, int iTimeWindow, bool  
16 fForceLogin, string strCoBrandedArgs, int iLangID, bool fSecure, string  
17 strNameSpace, int iKPP, bool bUseSecureAuth);  
18 [C++] public: String\* LogoTag(String\* strReturnUrl, int iTimeWindow, bool  
19 fForceLogin, String\* strCoBrandedArgs, int iLangID, bool fSecure, String\*  
20 strNameSpace, int iKPP, bool bUseSecureAuth);  
21 [VB] Public Function LogoTag(ByVal strReturnUrl As String, ByVal  
22 iTimeWindow As Integer, ByVal fForceLogin As Boolean, ByVal  
23 strCoBrandedArgs As String, ByVal iLangID As Integer, ByVal fSecure As  
24 Boolean, ByVal strNameSpace As String, ByVal iKPP As Integer, ByVal  
25 bUseSecureAuth As Boolean) As String

```

1 [JScript] public function LogoTag(strReturnUrl : String, iTimeWindow : int,
2 fForceLogin : Boolean, strCoBrandedArgs : String, iLangID : int, fSecure :
3 Boolean, strNameSpace : String, iKPP : int, bUseSecureAuth : Boolean) : String;

```

#### 5 *Description*

6       Similar to LogoTag(), this method returns an HTML snippet for the  
7 Passport Logo for the current member

8       For additional details on the parameters for this method, see the Passport  
9 SDK documentation. Sets the URL of the location that the Login server should  
10 redirect members to after sign-in is complete. Specifies the interval during which  
11 members must have last signed in. Determines how iTimeWindow gets used.  
12 Specifies variables to be appended as query string variables to the URL of the  
13 participant's Co-branding Template script page. Specifies the language to be used  
14 for the Login page that is displayed to the member. Declares whether this method  
15 is being called from an HTTPS (SSL) page. Specifies the domain in which the  
16 Passport should be created. Specifies data collection policies for purposes of  
17 COPPA compliance. Declares whether the actual sign-in UI should be served  
18 HTTPS from the Passport domain authority.

#### 19 *LogoTag*

```

21 [C#] public string LogoTag(string strReturnUrl, int iTimeWindow, int
22 iForceLogin, string strCoBrandedArgs, int iLangID, int iSecure, string
23 strNameSpace, int iKPP, int iUseSecureAuth);

```

```

24 [C++] public: String* LogoTag(String* strReturnUrl, int iTimeWindow, int
25 iForceLogin, String* strCoBrandedArgs, int iLangID, int iSecure, String*

```

```

1 strNameSpace, int iKPP, int iUseSecureAuth);
2 [VB] Public Function LogoTag(ByVal strReturnUrl As String, ByVal
3 iTimeWindow As Integer, ByVal iForceLogin As Integer, ByVal
4 strCoBrandedArgs As String, ByVal iLangID As Integer, ByVal iSecure As
5 Integer, ByVal strNameSpace As String, ByVal iKPP As Integer, ByVal
6 iUseSecureAuth As Integer) As String
7 [JScript] public function LogoTag(strReturnUrl : String, iTimeWindow : int,
8 iForceLogin : int, strCoBrandedArgs : String, iLangID : int, iSecure : int,
9 strNameSpace : String, iKPP : int, iUseSecureAuth : int) : String;
10
11 Description
12
13     LogoTag2
14
15 [C#] public string LogoTag2();
16 [C++] public: String* LogoTag2();
17 [VB] Public Function LogoTag2() As String
18 [JScript] public function LogoTag2() : String; Returns an HTML fragment
19 containing an < img > tag for a Passport link. This is based on the current state of
20 the identity (already logged on, and so on).

```

## *Description*

Returns an HTML fragment containing an < **img** > tag for a Passport link. This is based on the current state of the identity (already logged on, and so on).

The link image displays either **Sign In** if no valid ticket cookie is detected, or **Sign Out** if a valid cookie ticket is detected. For more details, see **IPassportManager2::LogoTag2** in the Passport SDK documentation.

#### LogoTag2

```
[C#] public string LogoTag2(string strReturnUrl, int iTimeWindow, bool
fForceLogin, string strCoBrandedArgs, int iLangID, bool fSecure, string
strNameSpace, int iKPP, bool bUseSecureAuth);

[C++] public: String* LogoTag2(String* strReturnUrl, int iTimeWindow, bool
fForceLogin, String* strCoBrandedArgs, int iLangID, bool fSecure, String*
strNameSpace, int iKPP, bool bUseSecureAuth);

[VB] Public Function LogoTag2(ByVal strReturnUrl As String, ByVal
iTimeWindow As Integer, ByVal fForceLogin As Boolean, ByVal
strCoBrandedArgs As String, ByVal iLangID As Integer, ByVal fSecure As
Boolean, ByVal strNameSpace As String, ByVal iKPP As Integer, ByVal
bUseSecureAuth As Boolean) As String

[JScript] public function LogoTag2(strReturnUrl : String, iTimeWindow : int,
fForceLogin : Boolean, strCoBrandedArgs : String, iLangID : int, fSecure :
Boolean, strNameSpace : String, iKPP : int, bUseSecureAuth : Boolean) : String;
```

#### *Description*

Returns an HTML fragment containing an **<img>** tag for a Passport link. This is based on the current state of the identity (already logged on, and so on).

The link image displays either **Sign In** if no valid ticket cookie is detected, or **Sign Out** if a valid cookie ticket is detected. For more details, see

1 **IPassportManager2::LogoTag2** in the Passport SDK documentation. See  
2 Passport documentation for **IPassportManager2::LoginUser** . See Passport  
3 documentation for **IPassportManager2::LoginUser** . See Passport  
4 documentation for **IPassportManager2::LoginUser** . See Passport  
5 documentation for **IPassportManager2::LoginUser** . See Passport  
6 documentation for **IPassportManager2::LoginUser** . See Passport  
7 documentation for **IPassportManager2::LoginUser** . See Passport  
8 documentation for **IPassportManager2::LoginUser** .

### 9 LogoTag2

10  
11 [C#] public string LogoTag2(string strReturnUrl, int iTimeWindow, int  
12 iForceLogin, string strCoBrandedArgs, int iLangID, int iSecure, string  
13 strNameSpace, int iKPP, int iUseSecureAuth);

14 [C++] public: String\* LogoTag2(String\* strReturnUrl, int iTimeWindow, int  
15 iForceLogin, String\* strCoBrandedArgs, int iLangID, int iSecure, String\*  
16 strNameSpace, int iKPP, int iUseSecureAuth);

17 [VB] Public Function LogoTag2(ByVal strReturnUrl As String, ByVal  
18 iTimeWindow As Integer, ByVal iForceLogin As Integer, ByVal  
19 strCoBrandedArgs As String, ByVal iLangID As Integer, ByVal iSecure As  
20 Integer, ByVal strNameSpace As String, ByVal iKPP As Integer, ByVal  
21 iUseSecureAuth As Integer) As String

22 [JScript] public function LogoTag2(strReturnUrl : String, iTimeWindow : int,  
23 iForceLogin : int, strCoBrandedArgs : String, iLangID : int, iSecure : int,  
24 strNameSpace : String, iKPP : int, iUseSecureAuth : int) : String;

## Description

Returns an HTML fragment containing an `<img>` tag for a Passport link. This is based on the current state of the identity (already logged on, and so on).

The link image displays either **Sign In** if no valid ticket cookie is detected, or **Sign Out** if a valid cookie ticket is detected. For more details, see **IPassportManager2::LogoTag2** in the Passport SDK documentation. See Passport documentation for **IPassportManager2::LoginUser** . See Passport documentation for **IPassportManager2::LoginUser** . See Passport documentation for **IPassportManager2::LoginUser** . See Passport documentation for **IPassportManager2::LoginUser** . See Passport documentation for **IPassportManager2::LoginUser** . See Passport documentation for **IPassportManager2::LoginUser** .

## SignOut

```
[C#] public static void SignOut(string strSignOutDotGifFileName);  
[C++] public: static void SignOut(String* strSignOutDotGifFileName);  
[VB] Public Shared Sub SignOut(ByVal strSignOutDotGifFileName As String)  
[JScript] public static function SignOut(strSignOutDotGifFileName : String);
```

## Description

Logs off the given Passport member from the current session. For more information, see the Passport SDK documentation . The *SignOut* image to be used.

1      UrlAuthorizationModule class (System.Web.Security)

2      ToString

3  
4  
5      *Description*

6          Provides URL-based authorization services for allowing or denying access  
7      to specified resources. This class cannot be inherited.

8      UrlAuthorizationModule

9      *Example Syntax:*

10      ToString

11  
12      [C#] public UrlAuthorizationModule();

13      [C++] public: UrlAuthorizationModule();

14      [VB] Public Sub New()

15      [JScript] public function UrlAuthorizationModule();

16      Dispose

17  
18      [C#] public void Dispose();

19      [C++] public: \_\_sealed void Dispose();

20      [VB] NotOverridable Public Sub Dispose()

21      [JScript] public function Dispose();

22  
23      *Description*

24          Called by the HTTP runtime to dispose of the module.  
25



1        Call **System.Web.Security.UrlAuthorizationModule.Dispose** when you  
2 are finished using the **System.Web.Security.UrlAuthorizationModule** . The  
3 **System.Web.Security.UrlAuthorizationModule.Dispose** method leaves the  
4 **System.Web.Security.UrlAuthorizationModule** in an unusable state. After  
5 calling **System.Web.Security.UrlAuthorizationModule.Dispose** , you must  
6 release all references to the **System.Web.Security.UrlAuthorizationModule** so  
7 the memory it occupied can be reclaimed by garbage collection.

#### 8        **Init**

9  
10    [C#] public void Init(HttpApplication app);  
11    [C++] public: \_\_sealed void Init(HttpApplication\* app);  
12    [VB] NotOverridable Public Sub Init(ByVal app As HttpApplication)  
13    [JScript] public function Init(app : HttpApplication);

#### 15        *Description*

16        Initializes the module. The **Http application** .

17        **WindowsAuthenticationEventArgs** class (System.Web.Security)

18        **ToString**

#### 21        *Description*

22        Provides data for the **WindowsAuthentication** event.

23        This is the event argument passed to the

24        **WindowsAuthentication\_OnAuthenticate** event handler. Contains a

25        **WindowsIdentity** object and the **IPrincipal** object used for the context.

WindowsAuthenticationEventArgs

*Example Syntax:*

ToString

```
[C#] public WindowsAuthenticationEventArgs(WindowsIdentity identity,  
HttpContext context);
```

```
[C++] public: WindowsAuthenticationEventArgs(WindowsIdentity* identity,  
HttpContext* context);
```

```
[VB] Public Sub New(ByVal identity As WindowsIdentity, ByVal context As  
HttpContext)
```

```
[JScript] public function WindowsAuthenticationEventArgs(identity :  
WindowsIdentity, context : HttpContext);
```

#### *Description*

Initializes a newly created instance of the WindowsAuthenticationEventArgs Class. The windows identity object. The context for the event.

Context

ToString

```
[C#] public HttpContext Context {get;}
```

```
[C++] public: __property HttpContext* get_Context();
```

```
[VB] Public ReadOnly Property Context As HttpContext
```

```
[JScript] public function get Context() : HttpContext;
```

1  
2 *Description*

3     The HttpContext intrinsic (provides access to Request, Response, and User  
4 objects).

5     Identity

6     ToString

7  
8 [C#] public WindowsIdentity Identity {get;}

9 [C++] public: \_\_property WindowsIdentity\* get\_Identity();

10 [VB] Public ReadOnly Property Identity As WindowsIdentity

11 [JScript] public function get Identity() : WindowsIdentity;

12  
13 *Description*

14     An authenticated Windows identity.

15     User

16     ToString

17  
18 [C#] public IPrincipal User {get; set;}

19 [C++] public: \_\_property IPrincipal\* get\_User();public: \_\_property void  
20 set\_User(IPrincipal\*);

21 [VB] Public Property User As IPrincipal

22 [JScript] public function get User() : IPrincipal;public function set  
23 User(IPrincipal);

24  
25 *Description*

1 IPrincipal object to be associated with the request. The user object should  
2 be attached to the contextIf User is non null and Context.User is null, the  
3 WindowsAuthenticationModule will initialize Context.User with  
4 WindowsAuthenticationEventArgs.User.

5 WindowsAuthenticationEventHandler delegate (System.Web.Security)  
6 ToString

7  
8  
9 *Description*

10 Represents the method that handles the  
11 **WindowsAuthentication\_OnAuthenticate** event of a  
12 **WindowsAuthenticationModule** . The source of the event. A  
13 **WindowsAuthenticationEventArgs** that contains the event data.

14 When you create a  
15 **System.Web.Security.WindowsAuthenticationEventHandler** delegate, you  
16 identify the method to handle the event. To associate the event with your  
17 EventHandler, add an instance of the delegate to the event. The EventHandler is  
18 called whenever the event occurs, unless you remove the delegate. For more  
19 information about EventHandler delegates, see .

20 WindowsAuthenticationModule class (System.Web.Security)  
21 ToString

22  
23  
24 *Description*  
25

1 Enables ASP.NET applications to use Windows/IIS authentication. This  
2 class cannot be inherited.

3 WindowsAuthenticationModule

4 *Example Syntax:*

5 ToString

6  
7 [C#] public WindowsAuthenticationModule();

8 [C++] public: WindowsAuthenticationModule();

9 [VB] Public Sub New()

10 [JScript] public function WindowsAuthenticationModule();

11 ToString

12  
13 [C#] public event WindowsAuthenticationEventHandler Authenticate;

14 [C++] public: \_\_event WindowsAuthenticationEventHandler\* Authenticate;

15 [VB] Public Event Authenticate As WindowsAuthenticationEventHandler

16  
17 *Description*

18 Raised during authentication. This is a Global.asax event that must be  
19 named **PassportAuthentication\_OnAuthenticate** event. It is used primarily to  
20 attach a custom **IPrincipal** object to the context.

21 For more information about handling events, see .

22 Dispose

23  
24 [C#] public void Dispose();

25 [C++] public: \_\_sealed void Dispose();

1 [VB] NotOverridable Public Sub Dispose()

2 [JScript] public function Dispose();

3  
4 *Description*

5 Disposes of the module derived from **IHttpModule** when called by the  
6 **HttpRuntime** .

7 Call **System.Web.Security.WindowsAuthenticationModule.Dispose**  
8 when you are finished using the  
9 **System.Web.Security.WindowsAuthenticationModule** . The  
10 **System.Web.Security.WindowsAuthenticationModule.Dispose** method leaves  
11 the **System.Web.Security.WindowsAuthenticationModule** in an unusable state.  
12 After calling **System.Web.Security.WindowsAuthenticationModule.Dispose** ,  
13 you must release all references to the  
14 **System.Web.Security.WindowsAuthenticationModule** so the memory it  
15 occupied can be reclaimed by garbage collection.

16 **Init**

17  
18 [C#] public void Init(HttpApplication app);

19 [C++] public: \_\_sealed void Init(HttpApplication\* app);

20 [VB] NotOverridable Public Sub Init(ByVal app As HttpApplication)

21 [JScript] public function Init(app : HttpApplication);

22  
23 *Description*

24 Initializes the module derived from **IHttpModule** when cal  
25

## System.Web.Services

### *Description*

The **System.Web.Services** namespace consists of the classes that enable you to build and use Web Services. A Web Service is a programmable entity residing on a Web Server exposed via standard Internet protocols.

WebMethodAttribute class (System.Web.Services)

### *Description*

Adding this attribute to a method within an ASP.NET Web Service makes the method callable from remote Web clients. This class cannot be inherited.

Methods within a class that have this attribute set are called Web Service methods. The method and class must be public and running inside an ASP.NET Web application.

Constructors:

WebMethodAttribute

*Example Syntax:*

[C#] public WebMethodAttribute();

[C++] public: WebMethodAttribute();

[VB] Public Sub New()

[JScript] public function WebMethodAttribute(); Initializes a new instance of the

1 **System.Web.Services.WebMethodAttribute** class.

2  
3 *Description*

4       Initializes a new instance of the  
5 **System.Web.Services.WebMethodAttribute** class.

6       WebMethodAttribute

7       *Example Syntax:*

8  
9 [C#] public WebMethodAttribute(bool enableSession);

10 [C++] public: WebMethodAttribute(bool enableSession);

11 [VB] Public Sub New(ByVal enableSession As Boolean)

12 [JScript] public function WebMethodAttribute(enableSession : Boolean);

13  
14 *Description*

15       Initializes a new instance of the  
16 **System.Web.Services.WebMethodAttribute** class. Initializes whether session  
17 state is enabled for the Web Service method.

18       WebMethodAttribute

19       *Example Syntax:*

20  
21 [C#] public WebMethodAttribute(bool enableSession, TransactionOption  
22 transactionOption);

23 [C++] public: WebMethodAttribute(bool enableSession, TransactionOption  
24 transactionOption);

25 [VB] Public Sub New(ByVal enableSession As Boolean, ByVal transactionOption



1 As TransactionOption)

2 [JScript] public function WebMethodAttribute(enableSession : Boolean,  
3 transactionOption : TransactionOption);

4  
5 *Description*

6       Initializes a new instance of the  
7 **System.Web.Services.WebMethodAttribute** class. Initializes whether session  
8 state is enabled for the Web Service method. Initializes the transaction support of a  
9 Web Service method.

10       WebMethodAttribute

11       *Example Syntax:*

12  
13 [C#] public WebMethodAttribute(bool enableSession, TransactionOption  
14 transactionOption, int cacheDuration);

15 [C++] public: WebMethodAttribute(bool enableSession, TransactionOption  
16 transactionOption, int cacheDuration);

17 [VB] Public Sub New(ByVal enableSession As Boolean, ByVal transactionOption  
18 As TransactionOption, ByVal cacheDuration As Integer)

19 [JScript] public function WebMethodAttribute(enableSession : Boolean,  
20 transactionOption : TransactionOption, cacheDuration : int);

21  
22 *Description*

23       Initializes a new instance of the  
24 **System.Web.Services.WebMethodAttribute** class. Initializes whether session

1 state is enabled for the Web Service method. Initializes the transaction support of a  
2 Web Service method. Initializes the number of seconds the response is cached.

3 WebMethodAttribute

4 *Example Syntax:*

5  
6 [C#] public WebMethodAttribute(bool enableSession, TransactionOption  
7 transactionOption, int cacheDuration, bool bufferResponse);

8 [C++] public: WebMethodAttribute(bool enableSession, TransactionOption  
9 transactionOption, int cacheDuration, bool bufferResponse);

10 [VB] Public Sub New(ByVal enableSession As Boolean, ByVal transactionOption  
11 As TransactionOption, ByVal cacheDuration As Integer, ByVal bufferResponse  
12 As Boolean)

13 [JScript] public function WebMethodAttribute(enableSession : Boolean,  
14 transactionOption : TransactionOption, cacheDuration : int, bufferResponse :  
15 Boolean);

16  
17 *Description*

18 Initializes a new instance of the  
19 **System.Web.Services.WebMethodAttribute** class. Initializes whether session  
20 state is enabled for the Web Service method. Initializes the transaction support of a  
21 Web Service method. Initializes the number of seconds the response is cached.  
22 Initializes whether the response for this request is buffered.

23 Properties:

24 BufferResponse  
25

1  
2 [C#] public bool BufferResponse {get; set;}

3 [C++] public: \_\_property bool get \_BufferResponse();public: \_\_property void  
4 set \_BufferResponse(bool);

5 [VB] Public Property BufferResponse As Boolean

6 [JScript] public function get BufferResponse() : Boolean;public function set  
7 BufferResponse(Boolean);

8  
9 *Description*

10 Gets or sets whether the response for this request is buffered.

11 Setting **System.Web.Services.WebMethodAttribute.BufferResponse** to  
12 **true** , serializes the response of the Web Service method into a memory buffer  
13 until either the response is completely serialized or the buffer is full. Once the  
14 response is buffered, it is returned to the Web Service client over the network.

15 When **System.Web.Services.WebMethodAttribute.BufferResponse** is **false** ,  
16 the response to the Web Service method is sent back to the client as it is serialized.

17 In general, you only want to set

18 **System.Web.Services.WebMethodAttribute.BufferResponse** to **false** , if it is  
19 known that a Web Service method returns large amounts of data to the client. For  
20 smaller amounts of data, Web Service performance is better with  
21 **System.Web.Services.WebMethodAttribute.BufferResponse** to **true** .

22 **CacheDuration**

23  
24 [C#] public int CacheDuration {get; set;}

25 [C++] public: \_\_property int get \_CacheDuration();public: \_\_property void

1 set\_CacheDuration(int);

2 [VB] Public Property CacheDuration As Integer

3 [JScript] public function get CacheDuration() : int;public function set

4 CacheDuration(int);

5  
6 *Description*

7 Gets or sets the number of seconds the response should be held in the  
8 cache.

9 When caching is enabled requests and responses are held in memory on the  
10 server for at least the cache duration so caution must be used if you expect  
11 requests or responses to be very large or you expect requests to vary widely.

12 *Description*

13  
14 [C#] public string Description {get; set;}

15 [C++] public: \_\_property String\* get\_Description();public: \_\_property void  
16 set\_Description(String\*);

17 [VB] Public Property Description As String

18 [JScript] public function get Description() : String;public function set

19 Description(String);

20  
21 *Description*

22 A descriptive message describing the Web Service method.

23 The descriptive message is displayed to prospective consumers of the Web  
24 Service when description documents for the Web Service are generated, such as  
25 the Service Description and the Service help page.

## EnableSession

[C#] public bool EnableSession {get; set;}

[C++] public: \_\_property bool get\_EnableSession();public: \_\_property void  
set\_EnableSession(bool);

[VB] Public Property EnableSession As Boolean

[JScript] public function get EnableSession() : Boolean;public function set  
EnableSession(Boolean);

### *Description*

Indicates whether session state is enabled for a Web Service method.

In order to store session state in the ASP.NET

**System.Web.SessionState.HttpSessionState** object, the Web Service must inherit  
from **System.Web.Services.WebService** and a

**System.Web.Services.WebMethodAttribute** applied to the Web Service method,  
setting the **System.Web.Services.WebMethodAttribute.EnableSession** property  
to **true** . If session state is not needed for a Web Service method, then disabling it  
may improve performance.

## MessageName

[C#] public string MessageName {get; set;}

[C++] public: \_\_property String\* get\_MessageName();public: \_\_property void  
set\_MessageName(String\*);

[VB] Public Property MessageName As String

[JScript] public function get MessageName() : String;public function set

1 MessageName(String);

2  
3 *Description*

4 The name used for the Web Service method in the data passed to and  
5 returned from a Web Service method.

6 The **System.Web.Services.WebMethodAttribute.MessageName** property  
7 can be used to alias method or property names. The most common use of the  
8 **System.Web.Services.WebMethodAttribute.MessageName** property will be to  
9 uniquely identify polymorphic methods. By default,

10 **System.Web.Services.WebMethodAttribute.MessageName** is set to the name  
11 of the Web Service method. Therefore, if a Web Service contains two or more  
12 Web Service methods with the same name, you can uniquely identify the  
13 individual Web Service methods by setting the  
14 **System.Web.Services.WebMethodAttribute.MessageName** to a name unique  
15 within the Web Service, without changing the name of the actual method name in  
16 code.

17 TransactionOption

18  
19 [C#] public TransactionOption TransactionOption {get; set;}

20 [C++] public: \_\_property TransactionOption get\_TransactionOption();public:  
21 \_\_property void set\_TransactionOption(TransactionOption);

22 [VB] Public Property TransactionOption As TransactionOption

23 [JScript] public function get TransactionOption() : TransactionOption;public  
24 function set TransactionOption(TransactionOption);

1  
2 *Description*

3 Indicates the transaction support of a Web Service method.

4 Web Service methods can only participate as the root object in a  
5 transaction, due to the stateless nature of the HTTP protocol. Web Service  
6 methods can invoke COM objects that participate in the same transaction as the  
7 Web Service method, if the COM object is marked to run within a transaction in  
8 the Component Services administrative tool. If a Web Service method, with a  
9 **System.Web.Services.WebMethodAttribute.TransactionOption** property of  
10 **Required** or **RequiresNew** invokes another Web Service method with a  
11 **System.Web.Services.WebMethodAttribute.TransactionOption** property of  
12 **Required** or **RequiresNew** , each Web Service method participates in their own  
13 transaction, because a Web Service method can only act as the root object in a  
14 transaction.

15 **TypeId**

16 **Methods:**

17 **WebService** class (**System.Web.Services**)

18 **ToString**

19  
20  
21 *Description*

22 Defines the optional base class for Web Services, which provides direct  
23 access to common ASP.NET objects, like those for application and session state.

24 If you don't need access to the common ASP.NET objects, you can still  
25 create a Web Service without deriving from **System.Web.Services.WebService** .

1 Additional ASP.NET objects can be accessed through

2 **System.Web.Services.WebService.Context** .

3     WebService

4     *Example Syntax:*

5     ToString

6  
7 [C#] public WebService();

8 [C++] public: WebService();

9 [VB] Public Sub New()

10 [JScript] public function WebService();

11     Application

12     ToString

13  
14 [C#] public HttpApplicationState Application {get;}

15 [C++] public: \_\_property HttpApplicationState\* get\_Application();

16 [VB] Public ReadOnly Property Application As HttpApplicationState

17 [JScript] public function get Application() : HttpApplicationState;

18  
19 *Description*

20     Gets the application object for the current HTTP request.

21     Web Services can utilize both application state and session state.

22     Application state is maintained across all sessions accessing a Web Service

23     regardless of whether session state is turned off for a method (by using the

24     **System.Web.Services.WebMethodAttribute.EnableSession** property of the

25     **System.Web.Services.WebMethodAttribute** ).



1 Container

2 Context

3 ToString

4  
5  
6 *Description*

7 Gets the ASP.NET **System.Web.HttpContext** for the current request,  
8 which encapsulates all HTTP-specific context used by the HTTP server to process  
9 Web requests.

10 If a Web Service method is one-way, then the  
11 **System.Web.Services.WebService.Context** property is always **null** . A Web  
12 Service is one-way when a  
13 **System.Web.Services.Protocols.SoapDocumentMethodAttribute** or  
14 **System.Web.Services.Protocols.SoapRpcMethodAttribute** is applied to a Web  
15 Service method and the  
16 **System.Web.Services.Protocols.SoapDocumentMethodAttribute.OneWay**  
17 property is set to **true** .

18 DesignMode

19 Events

20 Server

21 ToString

22  
23  
24 *Description*

25 Gets the **System.Web.HttpServerUtility** for the current request.

1       The **System.Web.HttpServerUtility** class provides several methods that  
2 can be used in the processing of Web requests, including  
3 **System.Web.HttpServerUtility.CreateObject(System.String)** (for instantiating  
4 COM objects).

5       Session

6       ToString

7  
8 [C#] public HttpSessionState Session {get;}

9 [C++] public: \_\_property HttpSessionState\* get\_Session();

10 [VB] Public ReadOnly Property Session As HttpSessionState

11 [JScript] public function get Session() : HttpSessionState;

12  
13 *Description*

14       Gets the **System.Web.SessionState.HttpSessionState** instance for the  
15 current request.

16       If a Web Service method is one-way, then the  
17 **System.Web.Services.WebService.Session** property is always **null** . A Web  
18 Service is one-way when a  
19 **System.Web.Services.Protocols.SoapDocumentMethodAttribute** or  
20 **System.Web.Services.Protocols.SoapRpcMethodAttribute** is applied to a Web  
21 Service method and the  
22 **System.Web.Services.Protocols.SoapDocumentMethodAttribute.OneWay**  
23 property is set to **true** .

24       Site

25       User

1 ToString

2  
3  
4 *Description*

5 Gets the ASP.NET server **System.Web.HttpContext.User** object. Can be  
6 used to authenticate whether a user is authorized to execute the request.

7 Both Internet Information Services (IIS) and the .NET Framework need to  
8 be configured for authentication in order for the

9 **System.Web.Services.WebService.User** property to be meaningful.

10 Authentication is the process of accepting credentials from a user and validating  
11 those credentials against some authority. If the credentials are valid, you have an  
12 authenticated identity. Authentication in the .NET Framework is configured by  
13 placing entries in the web.config file. Placing the XML in the following code into  
14 a web.config file sets the authentication mode to Windows.

15 WebServiceAttribute class (System.Web.Services)

16 ToString

17  
18  
19 *Description*

20 Used to add additional information to a Web Service, such as a string  
21 describing its functionality.

22 The **System.Web.Services.WebServiceAttribute** is not required for a  
23 Web Service to be published and executed. You can use the  
24 **WebServiceAttribute** to specify a name for the Web Service that is not restricted

1 by the rules for a Common Language Runtime identifier, which is used in the  
2 Service Description and the Service help page for the Web Service.

3 ToString

4  
5 [C#] public const string DefaultNamespace;  
6 [C++] public: const String\* DefaultNamespace;  
7 [VB] Public Const DefaultNamespace As String  
8 [JScript] public var DefaultNamespace : String;  
9

10 *Description*

11 The default value for the  
12 **System.Web.Services.WebServiceAttribute.Namespace** property. This field is  
13 constant.

14 WebServiceAttribute

15 *Example Syntax:*

16 ToString

17  
18 [C#] public WebServiceAttribute();  
19 [C++] public: WebServiceAttribute();  
20 [VB] Public Sub New()  
21 [JScript] public function WebServiceAttribute();  
22

23 *Description*

24 Initializes a new instance of the  
25 **System.Web.Services.WebServiceAttribute** class.

Description

ToString

[C#] public string Description {get; set;}

[C++] public: \_\_property String\* get\_Description();public: \_\_property void  
set\_Description(String\*);

[VB] Public Property Description As String

[JScript] public function get Description() : String;public function set  
Description(String);

### *Description*

A descriptive message for the Web Service.

The descriptive message is displayed to prospective consumers of the Web Service when description documents for the Web Service are generated, such as the Service Description and the Service help page.

Name

ToString

[C#] public string Name {get; set;}

[C++] public: \_\_property String\* get\_Name();public: \_\_property void  
set\_Name(String\*);

[VB] Public Property Name As String

[JScript] public function get Name() : String;public function set Name(String);

### *Description*

1 Gets or sets the name of the ASP.NET Web Service.

2 The Service Description is generated when a user navigates to the URL for  
3 the Web Service and supplies a query string of ?WSDL. Within the Service  
4 Description, the **System.Web.Services.WebServiceAttribute.Name** property  
5 identifies the local part of the XML qualified name for the Web Service. The  
6 **Name** property is also used to display the name of the Web Service on the Service  
7 help page. The Service help page is displayed when a prospective consumer  
8 navigates to the .asmx page for the Web Service without specifying a Web Service  
9 method name and its parameters.

10 Namespace

11 ToString

12  
13 [C#] public string Namespace {get; set;}

14 [C++] public: \_\_property String\* get\_Namespace();public: \_\_property void  
15 set\_Namespace(String\*);

16 [VB] Public Property Namespace As String

17 [JScript] public function get Namespace() : String;public function set  
18 Namespace(String);

19  
20 *Description*

21 Gets or sets the default XML namespace to use for the Web Service.

22 XML namespaces offer a way to create names in an XML document that  
23 are identified by a Uniform Resource Identifier (URI). By using XML namespaces  
24 you can uniquely identify elements or attributes in a XML document. The Service  
25

1 Description for a Web Service is defined in XML, specifically in Web Service  
2 Description Language (WSDL).

3       TypeId

4       WebServiceBindingAttribute class (System.Web.Services)

5       ToString

6  
7  
8 *Description*

9       Declares the binding one or more Web Service methods implemented  
10       within the class implementing the Web Service. This class cannot be inherited.

11       A binding, as defined by Web Services Description Language(WSDL), is  
12       similar to an interface, in that it defines a concrete set of operations. Each Web  
13       Service method is an operation within a particular binding. Web Service methods  
14       are members of either the default binding for a Web Service or in a binding  
15       specified within a **System.Web.Services.WebServiceBindingAttribute** applied  
16       to a class implementing a Web Service. A Web Service can implement multiple  
17       bindings, by applying multiple

18       **System.Web.Services.WebServiceBindingAttribute** attributes to a Web Service.

19       WebServiceBindingAttribute

20       *Example Syntax:*

21       ToString

22  
23       [C#] public WebServiceBindingAttribute();

24       [C++] public: WebServiceBindingAttribute();

25       [VB] Public Sub New()

1 [JScript] public function WebServiceBindingAttribute(); Initializes a new instance  
2 of the **System.Web.Services.WebServiceBindingAttribute** class.

3  
4 *Description*

5       Initializes a new instance of the  
6 **System.Web.Services.WebServiceBindingAttribute** class.

7       WebServiceBindingAttribute

8       *Example Syntax:*

9       ToString

10  
11 [C#] public WebServiceBindingAttribute(string name);

12 [C++] public: WebServiceBindingAttribute(String\* name);

13 [VB] Public Sub New(ByVal name As String)

14 [JScript] public function WebServiceBindingAttribute(name : String);

15  
16 *Description*

17       Initializes a new instance of the  
18 **System.Web.Services.WebServiceBindingAttribute** class setting the name of  
19 the binding the Web Service method is implementing.

20       This constructor is used to specify a name for a binding defined in the Web  
21 Service it is applied to and is a member of the default namespace. The name of the  
22 binding a Web Service method is implementing an operation for. Sets the  
23 **System.Web.Services.WebServiceBindingAttribute.Name** property.

24       WebServiceBindingAttribute

25       *Example Syntax:*



ToString

```
[C#] public WebServiceBindingAttribute(string name, string ns);  
[C++] public: WebServiceBindingAttribute(String* name, String* ns);  
[VB] Public Sub New(ByVal name As String, ByVal ns As String)  
[JScript] public function WebServiceBindingAttribute(name : String, ns : String);
```

### *Description*

Initializes a new instance of the

**System.Web.Services.WebServiceBindingAttribute** class.

This constructor is used to specify a name for a binding defined in the Web Service it is applied to that and is a member of the supplied namespace. The name of the binding a Web Service method is implementing an operation for. Sets the

**System.Web.Services.WebServiceBindingAttribute.Name** property. The

namespace associated with the binding. Sets the

**System.Web.Services.WebServiceBindingAttribute.Namespace** property.

WebServiceBindingAttribute

### *Example Syntax:*

ToString

```
[C#] public WebServiceBindingAttribute(string name, string ns, string location);  
[C++] public: WebServiceBindingAttribute(String* name, String* ns, String*  
location);  
[VB] Public Sub New(ByVal name As String, ByVal ns As String, ByVal location  
As String)
```

1 [JScript] public function WebServiceBindingAttribute(name : String, ns : String,  
2 location : String);

3  
4 *Description*

5       Initializes a new instance of the  
6 **System.Web.Services.WebServiceBindingAttribute** class.

7       This constructor is used to specify a name for a binding not defined in the  
8 Web Service it is applied to that and is a member of the supplied namespace. The  
9 name of the binding a Web Service method is implementing an operation for. Sets  
10 the **System.Web.Services.WebServiceBindingAttribute.Name** property. The  
11 namespace associated with the binding. Sets the  
12 **System.Web.Services.WebServiceBindingAttribute.Namespace** property. The  
13 location where the binding is defined.

14       Location

15       ToString

16  
17 [C#] public string Location {get; set;}

18 [C++] public: \_\_property String\* get\_Location();public: \_\_property void  
19 set\_Location(String\*);

20 [VB] Public Property Location As String

21 [JScript] public function get Location() : String;public function set  
22 Location(String);

23  
24 *Description*

25       Gets or sets the location where the binding is defined.

1 Name

2 ToString

3  
4 [C#] public string Name {get; set;}

5 [C++] public: \_\_property String\* get\_Name();pub

6  
7  
8  
9 **System.Web.Services.Configuration**

10  
11 *Description*

12  
13 WebServicesConfigurationSectionHandler class  
14 (System.Web.Services.Configuration)

15  
16  
17 *Description*

18 The configuration section handler for the webServices section of the  
19 Config.Web configuration file. The section handler participates in the resolution of  
20 configuration settings within the webServices portion of a Config.Web.

21 **Constructors:**

22 WebServicesConfigurationSectionHandler

23 *Example Syntax:*

24  
25 [C#] public WebServicesConfigurationSectionHandler();

```

1 [C++] public: WebServicesConfigurationSectionHandler();
2 [VB] Public Sub New()
3 [JScript] public function WebServicesConfigurationSectionHandler();
4     Methods:
5     Create
6
7 [C#] public object Create(object parent, object configContext, XmlNode section);
8 [C++] public: __sealed Object* Create(Object* parent, Object* configContext,
9 XmlNode* section);
10 [VB] NotOverridable Public Function Create(ByVal parent As Object, ByVal
11 configContext As Object, ByVal section As XmlNode) As Object
12 [JScript] public function Create(parent : Object, configContext : Object, section :
13 XmlNode) : Object;
14

```

### *Description*

Parses the configuration settings for the webServices portion of a Web.config configuration file to populate the values of a WebServicesConfiguration object and returning it. Reference to the "default" value provided by the parent IConfigurationSectionHandler. Provides access to the raw XML contents within a configuration file. The virtual path for which the configuration section handler should compute values.

XmlFormatExtensionAttribute class (System.Web.Services.Configuration)  
ToString

1  
2  
3 *Description*

4  
5 XmlFormatExtensionAttribute

6 *Example Syntax:*

7 ToString

8  
9 [C#] public XmlFormatExtensionAttribute();

10 [C++] public: XmlFormatExtensionAttribute();

11 [VB] Public Sub New()

12 [JScript] public function XmlFormatExtensionAttribute();

13  
14 *Description*

15  
16 XmlFormatExtensionAttribute

17 *Example Syntax:*

18 ToString

19  
20 [C#] public XmlFormatExtensionAttribute(string elementName, string ns, Type  
21 extensionPoint1);

22 [C++] public: XmlFormatExtensionAttribute(String\* elementName, String\* ns,  
23 Type\* extensionPoint1);

24 [VB] Public Sub New(ByVal elementName As String, ByVal ns As String, ByVal  
25 extensionPoint1 As Type)

1 [JScript] public function XmlFormatExtensionAttribute(elementName : String, ns  
2 : String, extensionPoint1 : Type);

3 XmlFormatExtensionAttribute

4 *Example Syntax:*

5 ToString

6  
7 [C#] public XmlFormatExtensionAttribute(string elementName, string ns, Type[]  
8 extensionPoints);

9 [C++] public: XmlFormatExtensionAttribute(String\* elementName, String\* ns,  
10 Type\* extensionPoints[]);

11 [VB] Public Sub New(ByVal elementName As String, ByVal ns As String, ByVal  
12 extensionPoints() As Type)

13 [JScript] public function XmlFormatExtensionAttribute(elementName : String, ns  
14 : String, extensionPoints : Type[]);

15  
16 *Description*

17  
18 XmlFormatExtensionAttribute

19 *Example Syntax:*

20 ToString

21  
22 [C#] public XmlFormatExtensionAttribute(string elementName, string ns, Type  
23 extensionPoint1, Type extensionPoint2);

24 [C++] public: XmlFormatExtensionAttribute(String\* elementName, String\* ns,  
25 Type\* extensionPoint1, Type\* extensionPoint2);

1 [VB] Public Sub New(ByVal elementName As String, ByVal ns As String, ByVal  
2 extensionPoint1 As Type, ByVal extensionPoint2 As Type)

3 [JScript] public function XmlFormatExtensionAttribute(elementName : String, ns  
4 : String, extensionPoint1 : Type, extensionPoint2 : Type);

5 XmlFormatExtensionAttribute

6 *Example Syntax:*

7 ToString

8  
9 [C#] public XmlFormatExtensionAttribute(string elementName, string ns, Type  
10 extensionPoint1, Type extensionPoint2, Type extensionPoint3);

11 [C++] public: XmlFormatExtensionAttribute(String\* elementName, String\* ns,  
12 Type\* extensionPoint1, Type\* extensionPoint2, Type\* extensionPoint3);

13 [VB] Public Sub New(ByVal elementName As String, ByVal ns As String, ByVal  
14 extensionPoint1 As Type, ByVal extensionPoint2 As Type, ByVal  
15 extensionPoint3 As Type)

16 [JScript] public function XmlFormatExtensionAttribute(elementName : String, ns  
17 : String, extensionPoint1 : Type, extensionPoint2 : Type, extensionPoint3 : Type);

18 XmlFormatExtensionAttribute

19 *Example Syntax:*

20 ToString

21  
22 [C#] public XmlFormatExtensionAttribute(string elementName, string ns, Type  
23 extensionPoint1, Type extensionPoint2, Type extensionPoint3, Type  
24 extensionPoint4);

25 [C++] public: XmlFormatExtensionAttribute(String\* elementName, String\* ns,

```

1  Type* extensionPoint1, Type* extensionPoint2, Type* extensionPoint3, Type*
2  extensionPoint4);
3  [VB] Public Sub New(ByVal elementName As String, ByVal ns As String, ByVal
4  extensionPoint1 As Type, ByVal extensionPoint2 As Type, ByVal
5  extensionPoint3 As Type, ByVal extensionPoint4 As Type)
6  [JScript] public function XmlFormatExtensionAttribute(elementName : String, ns
7  : String, extensionPoint1 : Type, extensionPoint2 : Type, extensionPoint3 : Type,
8  extensionPoint4 : Type);
9      Properties:
10     ElementName
11     ToString
12
13 [C#] public string ElementName {get; set;}
14 [C++] public: __property String* get_ElementName();public: __property void
15 set_ElementName(String*);
16 [VB] Public Property ElementName As String
17 [JScript] public function get ElementName() : String;public function set
18 ElementName(String);
19
20 Description
21
22     ExtensionPoints
23     ToString
24
25 [C#] public Type[] ExtensionPoints {get; set;}

```



```

1 [C++] public: __property Type* get_ExtensionPoints();public: __property void
2 set_ExtensionPoints(Type*[]);
3 [VB] Public Property ExtensionPoints As Type ()
4 [JScript] public function get ExtensionPoints() : Type[];public function set
5 ExtensionPoints(Type[]);
6

```

### Description

Namespace

ToString

```

12 [C#] public string Namespace {get; set;}

```

```

13 [C++] public: __property String* get_Namespace();public: __property void
14 set_Namespace(String*);

```

```

15 [VB] Public Property Namespace As String

```

```

16 [JScript] public function get Namespace() : String;public function set
17 Namespace(String);

```

### Description

TypeId

XmlFormatExtensionPointAttribute class

(System.Web.Services.Configuration)

ToString

1  
2  
3 *Description*  
4

5 XmlFormatExtensionPointAttribute

6 *Example Syntax:*

7 ToString

8  
9 [C#] public XmlFormatExtensionPointAttribute(string memberName);

10 [C++] public: XmlFormatExtensionPointAttribute(String\* memberName);

11 [VB] Public Sub New(ByVal memberName As String)

12 [JScript] public function XmlFormatExtensionPointAttribute(memberName :  
13 String);

14  
15 *Description*  
16

17 AllowElements

18 ToString

19  
20 [C#] public bool AllowElements {get; set;}

21 [C++] public: \_\_property bool get\_AllowElements();public: \_\_property void  
22 set\_AllowElements(bool);

23 [VB] Public Property AllowElements As Boolean

24 [JScript] public function get AllowElements() : Boolean;public function set  
25 AllowElements(Boolean);

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

*Description*

MemberName  
ToString

```
[C#] public string MemberName {get; set;}  
[C++] public: __property String* get_MemberName();public: __property void  
set_MemberName(String*);  
[VB] Public Property MemberName As String  
[JScript] public function get MemberName() : String;public function set  
MemberName(String);
```

*Description*

TypeId  
XmlFormatExtensionPrefixAttribute class  
(System.Web.Services.Configuration)  
ToString

*Description*

XmlFormatExtensionPrefixAttribute  
Example Syntax:

1 ToString

2  
3 [C#] public XmlFormatExtensionPrefixAttribute();

4 [C++] public: XmlFormatExtensionPrefixAttribute();

5 [VB] Public Sub New()

6 [JScript] public function XmlFormatExtensionPrefixAttribute();

7  
8 *Description*

9  
10 XmlFormatExtensionPrefixAttribute

11 *Example Syntax:*

12 ToString

13  
14 [C#] public XmlFormatExtensionPrefixAttribute(string prefix, string ns);

15 [C++] public: XmlFormatExtensionPrefixAttribute(String\* prefix, String\* ns);

16 [VB] Public Sub New(ByVal prefix As String, ByVal ns As String)

17 [JScript] public function XmlFormatExtensionPrefixAttribute(prefix : String, ns :  
18 String);

19  
20 *Description*

21  
22 Namespace

23 ToString

24  
25 [C#] public string Namespace {get; set;}

```

1 [C++] public: __property String* get_Namespace();public: __property void
2 set_Namespace(String*);
3 [VB] Public Property Namespace As String
4 [JScript] public function get Namespace() : String;public function set
5 Namespace(String);

```

### *Description*

Prefix

ToString

```

12 [C#] public string Prefix {get; set;}

```

```

13 [C++]

```

## **System.Web.Services.Description**

### *Description*

The **System.Web.Services.Description** namespace consists of the classes that enable you to publicly describe a Web Service by using the Web Service Description Language (WSDL). Each class in the **System.Web.Services.Description** namespace corresponds to a specific element in the WSDL specification, and the class hierarchy corresponds to the XML

1 structure of a valid WSDL document. For more information about WSDL, see the  
2 specification at <http://www.w3.org/TR/wsdl/>.

3 Binding class (System.Web.Services.Description)

4  
5  
6 *Description*

7 Specifies the concrete data format and protocols used in the Web Service.

8 This class cannot be inherited.

9 Constructors:

10 Binding

11 *Example Syntax:*

12  
13 [C#] public Binding();

14 [C++] public: Binding();

15 [VB] Public Sub New()

16 [JScript] public function Binding();

17 Properties:

18 Documentation

19 Extensions

20  
21  
22 *Description*

23 Gets the collection of extensibility elements used in the Web Service.

24 Name

1  
2 [C#] public string Name {get; set;}

3 [C++] public: \_\_property String\* get\_Name();public: \_\_property void

4 set\_Name(String\*);

5 [VB] Public Property Name As String

6 [JScript] public function get Name() : String;public function set Name(String);

7  
8 *Description*

9 Gets or sets a string value containing the name of the  
10 **System.Web.Services.Description.Binding** .

11 **Operations**

12  
13 [C#] public OperationBindingCollection Operations {get;}

14 [C++] public: \_\_property OperationBindingCollection\* get\_Operations();

15 [VB] Public ReadOnly Property Operations As OperationBindingCollection

16 [JScript] public function get Operations() : OperationBindingCollection;

17  
18 *Description*

19 Gets the collection of the specifications for data formats and message  
20 protocols used in the action supported by the Web Service.

21 **ServiceDescription**

22  
23 [C#] public ServiceDescription ServiceDescription {get;}

24 [C++] public: \_\_property ServiceDescription\* get\_ServiceDescription();

25 [VB] Public ReadOnly Property ServiceDescription As ServiceDescription

1 [JScript] public function get ServiceDescription() : ServiceDescription;

2  
3 *Description*

4 Gets the **System.Web.Services.Description.ServiceDescription** instance  
5 of which the **System.Web.Services.Description.Binding** is a member.

6 **Type**

7  
8 [C#] public XmlQualifiedName Type {get; set;}

9 [C++] public: \_\_property XmlQualifiedName\* get\_Type();public: \_\_property  
10 void set\_Type(XmlQualifiedName\*);

11 [VB] Public Property Type As XmlQualifiedName

12 [JScript] public function get Type() : XmlQualifiedName;public function set  
13 Type(XmlQualifiedName);

14  
15 *Description*

16 Gets or sets a value representing the XML datatype definitions used by the  
17 Web Service.

18 The default implementation is **String.Empty** .

19 **Methods:**

20 **BindingCollection** class (System.Web.Services.Description)

21 **ToString**

22  
23  
24 *Description*

25



1 Represents a collection of **System.Web.Services.Description.Binding**  
2 elements supported by the Web Service. This class cannot be inherited.

3 Count

4 InnerList

5 Item

6 ToString

7 **System.Web.Services.Description.Binding**

8  
9 *Description*

10 Gets or sets the value of a **System.Web.Services.Description.Binding** at  
11 the specified zero-based index. The zero-based index of the  
12 **System.Web.Services.Description.Binding** whose value is modified or returned.

13 Item

14 ToString

15  
16 [C#] public Binding this[string name] {get;}

17 [C++] public: \_\_property Binding\* get\_Item(String\* name);

18 [VB] Public Default ReadOnly Property Item(ByVal name As String) As Binding

19 [JScript] returnValue = BindingCollectionObject.Item(name);

20  
21 *Description*

22 Gets a **System.Web.Services.Description.Binding** specified by its **Name**  
23 property. A string value representing the name of the Binding returned.

24 List

25 Table

## Add

[C#] public int Add(Binding binding);

[C++] public: int Add(Binding\* binding);

[VB] Public Function Add(ByVal binding As Binding) As Integer

[JScript] public function Add(binding : Binding) : int;

### *Description*

Adds the specified **System.Web.Services.Description.Binding** to the end of the **System.Web.Services.Description.BindingCollection**.

*Return Value:* Returns the zero-based index where *binding* has been added. The **System.Web.Services.Description.Binding** to be added to the collection.

### Contains

[C#] public bool Contains(Binding binding);

[C++] public: bool Contains(Binding\* binding);

[VB] Public Function Contains(ByVal binding As Binding) As Boolean

[JScript] public function Contains(binding : Binding) : Boolean;

### *Description*

Gets a value indicating whether the specified **System.Web.Services.Description.Binding** is a member of the **System.Web.Services.Description.BindingCollection**.

*Return Value:* **true** if *binding* is a member of the

1 **System.Web.Services.Description.BindingCollection** ; otherwise, **false** . A

2 **System.Web.Services.Description.Binding** object.

3 **CopyTo**

4  
5 [C#] public void CopyTo(Binding[] array, int index);

6 [C++] public: void CopyTo(Binding\* array[], int index);

7 [VB] Public Sub CopyTo(ByVal array() As Binding, ByVal index As Integer)

8 [JScript] public function CopyTo(array : Binding[], index : int);

9  
10 *Description*

11 Copies the entire **System.Web.Services.Description.BindingCollection** to  
12 a compatible one-dimensional array of type

13 **System.Web.Services.Description.Binding** , starting at the specified zero-based  
14 index of the target array. An array of type

15 **System.Web.Services.Description.Binding** serving as the destination for the  
16 copy action. The zero-based index at which to start placing the copied collection.

17 **GetKey**

18  
19 [C#] protected override string GetKey(object value);

20 [C++] protected: String\* GetKey(Object\* value);

21 [VB] Overrides Protected Function GetKey(ByVal value As Object) As String

22 [JScript] protected override function GetKey(value : Object) : String;

23  
24 *Description*

25

1 Returns the name of the **System.Web.Services.Description.Binding**  
2 associated with the value passed by reference. An object for which to return the  
3 name of the key.

#### 4 IndexOf

5  
6 [C#] public int IndexOf(Binding binding);  
7 [C++] public: int IndexOf(Binding\* binding);  
8 [VB] Public Function IndexOf(ByVal binding As Binding) As Integer  
9 [JScript] public function IndexOf(binding : Binding) : int;

#### 11 *Description*

12 Searches for the specified **System.Web.Services.Description.Binding** and  
13 returns the zero-based index of the first occurrence within the collection.

14 *Return Value:* Returns a 32-bit signed integer. A  
15 **System.Web.Services.Description.Binding** object.

#### 16 Insert

17  
18 [C#] public void Insert(int index, Binding binding);  
19 [C++] public: void Insert(int index, Binding\* binding);  
20 [VB] Public Sub Insert(ByVal index As Integer, ByVal binding As Binding)  
21 [JScript] public function Insert(index : int, binding : Binding);

#### 23 *Description*

24 Adds the specified **System.Web.Services.Description.Binding** to the  
25 **System.Web.Services.Description.BindingCollection** at the specified index.

1        If the number of items in the collection already equals the collection's  
2 capacity, the capacity is doubled by automatically reallocating the internal array  
3 before the new element is inserted. The zero-based index at which to insert  
4 *binding*. The **System.Web.Services.Description.Binding** to be added to the  
5 collection.

#### 6        Remove

7  
8        [C#] public void Remove(Binding binding);  
9        [C++] public: void Remove(Binding\* binding);  
10        [VB] Public Sub Remove(ByVal binding As Binding)  
11        [JScript] public function Remove(binding : Binding);  
12

#### 13        *Description*

14        Removes the first occurrence of the specified  
15 **System.Web.Services.Description.Binding** from the  
16 **System.Web.Services.Description.BindingCollection** .

17        This method performs a linear search; therefore, the average execution time  
18 is proportional to **System.Web.Services.Description.BindingCollection.Count** .

19        A **System.Web.Services.Description.Binding** object.

#### 20        SetParent

21  
22        [C#] protected override void SetParent(object value, object parent);  
23        [C++] protected: void SetParent(Object\* value, Object\* parent);  
24        [VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As  
25        Object)

[JScript] protected override function SetParent(value : Object, parent : Object);

### *Description*

Sets the parent **System.Web.Services.Description.ServiceDescription** of a member of **System.Web.Services.Description.BindingCollection** . An object, of type **System.Web.Services.Description.Binding** , within the collection. The object, of type **System.Web.Services.Description.ServiceDescription** , to set as the parent.

DocumentableItem class (System.Web.Services.Description)

ToString

### *Description*

Represents the abstract base class from which several classes in the **System.Web.Services.Description** namespace are derived.

Several classes are derived from this class, including:

**System.Web.Services.Description.BindingSystem.Web.Services.Description.ImportSystem.Web.Services.Description.MessageSystem.Web.Services.Description.MessagePartSystem.Web.Services.Description.OperationSystem.Web.Services.Description.OperationBindingSystem.Web.Services.Description.OperationMessageSystem.Web.Services.Description.PortSystem.Web.Services.Description.PortTypeSystem.Web.Services.Description.ServiceSystem.Web.Services.Description.Types** This base class contains only one property, the **Documentation** property. It represents the text documentation for a specific XML element.

DocumentableItem

*Example Syntax:*

ToString

[C#] protected DocumentableItem();

[C++] protected: DocumentableItem();

[VB] Protected Sub New()

[JScript] protected function DocumentableItem();

Documentation

ToString

[C#] public string Documentation {get; set;}

[C++] public: \_\_property String\* get\_Documentation();public: \_\_property void

set\_Documentation(String\*);

[VB] Public Property Documentation As String

[JScript] public function get Documentation() : String;public function set

Documentation(String);

*Description*

Gets or sets the documentation for the instance of the  
**System.Web.Services.Description.DocumentableItem** .

In a derived class, this property represents the text comments added to an  
element of the Web Service. The default implementation is **String.Empty** .

FaultBinding class (System.Web.Services.Description)

ToString

1  
2  
3 *Description*

4 Specifies the format for any error messages that might be output as a result  
5 of the operation. This class cannot be inherited.

6 FaultBinding

7 *Example Syntax:*

8 ToString

9  
10 [C#] public FaultBinding();

11 [C++] public: FaultBinding();

12 [VB] Public Sub New()

13 [JScript] public function FaultBinding();

14 Documentation

15 Extensions

16 ToString

17  
18  
19 *Description*

20 Gets the collection of extensibility elements associated with the  
21 **System.Web.Services.Description.FaultBinding** .

22 Name

23 OperationBinding

24 FaultBindingCollection class (System.Web.Services.Description)

25 ToString



1  
2  
3 *Description*

4 Represents a collection of  
5 **System.Web.Services.Description.FaultBinding** instances. This class cannot be  
6 inherited.

7 Count

8 InnerList

9 Item

10 ToString

11 **System.Web.Services.Description.FaultBinding**

12  
13 *Description*

14 Gets or sets the value of a  
15 **System.Web.Services.Description.FaultBinding** at the specified zero-based  
16 index. The zero-based index of the  
17 **System.Web.Services.Description.FaultBinding** whose value is modified or  
18 returned.

19 Item

20 ToString

21  
22 [C#] public FaultBinding this[string name] {get;}

23 [C++] public: \_\_property FaultBinding\* get\_Item(String\* name);

24 [VB] Public Default ReadOnly Property Item(ByVal name As String) As

25 FaultBinding

1 [JScript] return Value = FaultBindingCollectionObject.Item(name);

3 *Description*

4 Gets a **System.Web.Services.Description.FaultBinding** specified by its  
5 **System.Web.Services.Description.FaultBinding.Name** property. The string  
6 value representing the name of the  
7 **System.Web.Services.Description.FaultBinding** returned.

8 List

9 Table

10 Add

12 [C#] public int Add(FaultBinding bindingOperationFault);

13 [C++] public: int Add(FaultBinding\* bindingOperationFault);

14 [VB] Public Function Add(ByVal bindingOperationFault As FaultBinding) As

15 Integer

16 [JScript] public function Add(bindingOperationFault : FaultBinding) : int;

18 *Description*

19 Adds the specified **System.Web.Services.Description.FaultBinding** to the  
20 end of the **System.Web.Services.Description.FaultBindingCollection** .

21 *Return Value:* Returns the index where *bindingOperationFault* has been added.

22 The **System.Web.Services.Description.FaultBinding** to be added to the  
23 collection.

24 Contains

```

1
2 [C#] public bool Contains(FaultBinding bindingOperationFault);
3 [C++] public: bool Contains(FaultBinding* bindingOperationFault);
4 [VB] Public Function Contains(ByVal bindingOperationFault As FaultBinding)
5 As Boolean
6 [JScript] public function Contains(bindingOperationFault : FaultBinding) :
7 Boolean;
8

```

### 9 *Description*

10 Gets a value indicating whether the specified

11 **System.Web.Services.Description.FaultBinding** is a member of the

12 **FaultBindingCollection** .

13 *Return Value:* **true** if *bindingOperationFault* is a member of the

14 **FaultBindingCollection** ; otherwise, **false** . A

15 **System.Web.Services.Description.FaultBinding** object.

### 16 *CopyTo*

```

17
18 [C#] public void CopyTo(FaultBinding[] array, int index);
19 [C++] public: void CopyTo(FaultBinding* array[], int index);
20 [VB] Public Sub CopyTo(ByVal array() As FaultBinding, ByVal index As
21 Integer)
22 [JScript] public function CopyTo(array : FaultBinding[], index : int);
23

```

### 24 *Description*

Copies the entire **FaultBindingCollection** to a compatible one-dimensional array of type **System.Web.Services.Description.FaultBinding** , starting at the specified zero-based index of the target array. An array of type **System.Web.Services.Description.FaultBinding** serving as the destination for the copy action. The zero-based index at which to start placing the copied collection.

#### GetKey

[C#] protected override string GetKey(object value);

[C++] protected: String\* GetKey(Object\* value);

[VB] Overrides Protected Function GetKey(ByVal value As Object) As String

[JScript] protected override function GetKey(value : Object) : String;

#### *Description*

Returns the name of the **System.Web.Services.Description.FaultBinding** associated with the value passed by reference. An object for which to return the name.

#### IndexOf

[C#] public int IndexOf(FaultBinding bindingOperationFault);

[C++] public: int IndexOf(FaultBinding\* bindingOperationFault);

[VB] Public Function IndexOf(ByVal bindingOperationFault As FaultBinding) As Integer

[JScript] public function IndexOf(bindingOperationFault : FaultBinding) : int;

1  
2 *Description*

3 Searches for the specified  
4 **System.Web.Services.Description.FaultBinding** and returns the zero-based  
5 index of the first occurrence within the collection.

6 *Return Value:* Returns a 32-bit signed integer. A  
7 **System.Web.Services.Description.FaultBinding** instance.

8 Insert

9  
10 [C#] public void Insert(int index, FaultBinding bindingOperationFault);  
11 [C++] public: void Insert(int index, FaultBinding\* bindingOperationFault);  
12 [VB] Public Sub Insert(ByVal index As Integer, ByVal bindingOperationFault As  
13 FaultBinding)  
14 [JScript] public function Insert(index : int, bindingOperationFault : FaultBinding);  
15

16 *Description*

17 Adds the specified **System.Web.Services.Description.FaultBinding** to the  
18 **FaultBindingCollection** at the specified zero-based index.

19 If the number of items in the collection already equals the collection's  
20 capacity, the capacity is doubled by automatically reallocating the internal array  
21 before the new element is inserted. The zero-based index at which to insert  
22 *bindingOperationFault*. The **System.Web.Services.Description.FaultBinding** to  
23 be added to the collection.

24 Remove

```

1
2 [C#] public void Remove(FaultBinding bindingOperationFault);
3 [C++] public: void Remove(FaultBinding* bindingOperationFault);
4 [VB] Public Sub Remove(ByVal bindingOperationFault As FaultBinding)
5 [JScript] public function Remove(bindingOperationFault : FaultBinding);
6

```

### *Description*

Removes the first occurrence the specified **System.Web.Services.Description.FaultBinding** from the **FaultBindingCollection**.

This method performs a linear search; therefore, the average execution time is proportional to

**System.Web.Services.Description.FaultBindingCollection.Count**. The **System.Web.Services.Description.FaultBinding** object to be removed from the collection.

### **SetParent**

```

17
18 [C#] protected override void SetParent(object value, object parent);
19 [C++] protected: void SetParent(Object* value, Object* parent);
20 [VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As
21 Object)
22 [JScript] protected override function SetParent(value : Object, parent : Object);
23

```

### *Description*

1       Sets the parent **System.Web.Services.Description.OperationBinding** of a  
2 member of the **FaultBindingCollection** instance. An object, of type  
3 **System.Web.Services.Description.FaultBinding**, within the  
4 **FaultBindingCollection**. The object, of type  
5 **System.Web.Services.Description.OperationBinding**, to set as the parent.

6       HttpAddressBinding class (System.Web.Services.Description)

7       ToString

8  
9  
10    *Description*

11       Represents an extensibility element added to a  
12 **System.Web.Services.Description.Port** within a Web Service. This class cannot  
13 be inherited.

14       This class, through its  
15 **System.Web.Services.Description.HttpAddressBinding.Location** property,  
16 specifies the base URI for the Web Service. For more information, see .

17       HttpAddressBinding

18       *Example Syntax:*

19       ToString

20  
21    [C#] public HttpAddressBinding();

22    [C++] public: HttpAddressBinding();

23    [VB] Public Sub New()

24    [JScript] public function HttpAddressBinding();

25       Handled

Location

ToString

*Description*

Gets or sets a value representing the URL of the .asmx file containing the methods exposed by the Web Service.

Parent

Required

HttpBinding class (System.Web.Services.Description)

ToString

*Description*

Represents an extensibility element added to a **System.Web.Services.Description.Binding** within a Web Service. This class cannot be inherited.

This class specifies that information is to be passed by means of HTTP. For more information about specification of protocols for Web Services, see .

ToString

[C#] public const string Namespace;

[C++] public: const String\* Namespace;

[VB] Public Const Namespace As String

[JScript] public var Namespace : String;



1  
2 *Description*

3 Specifies the URI (<http://schemas.xmlsoap.org/wsdl/http/>) for the XML  
4 namespace of the **HttpBinding** class.

5 **HttpBinding**

6 *Example Syntax:*

7 **ToString**

8  
9 **[C#] public HttpBinding();**

10 **[C++] public: HttpBinding();**

11 **[VB] Public Sub New()**

12 **[JScript] public function HttpBinding();**

13 **Handled**

14 **Parent**

15 **Required**

16 **Verb**

17 **ToString**

18  
19  
20 *Description*

21 Gets or sets a value indicating whether the Http request will be made using  
22 the "GET" or "POST" method.

23 **HttpOperationBinding** class (System.Web.Services.Description)

24 **ToString**

1  
2  
3 *Description*

4 Represents an extensibility element added to an  
5 **System.Web.Services.Description.OperationBinding** within a Web Service.

6 This class cannot be inherited.

7 This class specifies how a Web Service passes information by means of  
8 HTTP. For more information, see .

9 **HttpOperationBinding**

10 *Example Syntax:*

11 **ToString**

12  
13 [C#] public **HttpOperationBinding**();

14 [C++] public: **HttpOperationBinding**();

15 [VB] Public Sub New()

16 [JScript] public function **HttpOperationBinding**();

17 **Handled**

18 **Location**

19 **ToString**

20  
21  
22 *Description*

23 Gets or sets a value representing the relative URL, within the WSDL  
24 document, of the action supported by the **HttpOperationBinding** .

25 **Parent**

Required

HttpUrlEncodedBinding class (System.Web.Services.Description)

ToString

### *Description*

Represents an extensibility element added to an **System.Web.Services.Description.InputBinding** within a Web Service. This class cannot be inherited.

This class specifies that the incoming data will be in the format "name=value&name=value", encoded from an **System.Web.UI.HtmlControls.HtmlForm**, whether the value of the **HtmlForm's System.Web.UI.HtmlControls.HtmlForm.Method** property is **POST** or **GET**.

HttpUrlEncodedBinding

*Example Syntax:*

ToString

[C#] public HttpUrlEncodedBinding();

[C++] public: HttpUrlEncodedBinding();

[VB] Public Sub New()

[JScript] public function HttpUrlEncodedBinding();

Handled

Parent

Required

HttpUrlReplacementBinding class (System.Web.Services.Description)

ToString

*Description*

Represents an extensibility element added to an **System.Web.Services.Description.InputBinding** within a Web Service. It specifies the format for data transmission through HTTP. This class cannot be inherited.

This class allows users to specify a custom format for data transmission, rather than limiting them to the standard format, "name=value&name=value", used in HTTP query strings. For example, a developer may decide to use /name,value/name,value or some other custom format instead.

HttpRequestReplacementBinding

*Example Syntax:*

ToString

[C#] public HttpRequestReplacementBinding();

[C++] public: HttpRequestReplacementBinding();

[VB] Public Sub New()

[JScript] public function HttpRequestReplacementBinding();

Handled

Parent

Required

Import class (System.Web.Services.Description)

ToString

### *Description*

Associates an XML namespace with a document location. This class cannot be inherited.

Many of the tags in the WSDL schema can contain the tag. It allows the separation of different elements of the Web Service into different documents, which can then be imported as needed.

Import

*Example Syntax:*

ToString

[C#] public Import();

[C++] public: Import();

[VB] Public Sub New()

[JScript] public function Import();

Documentation

Location

ToString

### *Description*

Gets or sets a reference to the XML **Location** attribute of the **System.Web.Services.Description.Import** instance.

Namespace

ToString

[C#] public string Namespace {get; set;}

[C++] public: \_\_property String\* get\_Namespace();public: \_\_property void  
set\_Namespace(String\*);

[VB] Public Property Namespace As String

[JScript] public function get Namespace() : String;public function set  
Namespace(String);

#### *Description*

Gets or sets a reference to the XML Namespace of the  
**System.Web.Services.Description.Import** instance.

ServiceDescription

ToString

[C#] public ServiceDescription ServiceDescription {get;}

[C++] public: \_\_property ServiceDescription\* get\_ServiceDescription();

[VB] Public ReadOnly Property ServiceDescription As ServiceDescription

[JScript] public function get ServiceDescription() : ServiceDescription;

#### *Description*

Gets a reference to the  
**System.Web.Services.Description.ServiceDescription** of which the **Import**  
instance is a member.

ImportCollection class (System.Web.Services.Description)

ToString

*Description*

Provides a collection of **System.Web.Services.Description.Import** instances representing documents to be imported into the Web Service. This class cannot be inherited.

Count

InnerList

Item

ToString

*Description*

Gets or sets the value of an **System.Web.Services.Description.Import** at the specified zero-based index. The zero-based index of the **Import** whose value is modified or returned.

List

Table

Add

[C#] public int Add(Import import);

[C++] public: int Add(Import\* import);

[VB] Public Function Add(ByVal import As Import) As Integer

[JScript] public function Add(import : Import) : int;

## Description

Adds the specified **System.Web.Services.Description.Import** instance to the end of the **System.Web.Services.Description.ImportCollection** .

*Return Value:* Returns the index where *import* has been added. The **System.Web.Services.Description.Import** to be added to the collection.

## Contains

[C#] public bool Contains(Import import);

[C++] public: bool Contains(Import\* import);

[VB] Public Function Contains(ByVal import As Import) As Boolean

[JScript] public function Contains(import : Import) : Boolean;

## Description

Gets a value indicating whether the specified **System.Web.Services.Description.Import** instance is a member of the **System.Web.Services.Description.ImportCollection** .

*Return Value:* **true** if *import* is a member of the **System.Web.Services.Description.ImportCollection** ; otherwise, **false** . An **System.Web.Services.Description.Import** object.

## CopyTo

[C#] public void CopyTo(Import[] array, int index);

[C++] public: void CopyTo(Import\* array[], int index);

[VB] Public Sub CopyTo(ByVal array() As Import, ByVal index As Integer)



1 [JScript] public function CopyTo(array : Import[], index : int);

3 *Description*

4 Copies the entire **ImportCollection** to a compatible one-dimensional array  
5 of type **System.Web.Services.Description.Import** , starting at the specified zero-  
6 based index of the target array. The destination array, of type  
7 **System.Web.Services.Description.Import**. The zero-based index at which to  
8 start placing the copied collection.

9 **IndexOf**

11 [C#] public int IndexOf(Import import);

12 [C++] public: int IndexOf(Import\* import);

13 [VB] Public Function IndexOf(ByVal import As Import) As Integer

14 [JScript] public function IndexOf(import : Import) : int;

16 *Description*

17 Searches for the specified **System.Web.Services.Description.Import** and  
18 returns the zero-based index of the first occurrence within the collection.

19 *Return Value:* Returns a 32-bit signed integer. An

20 **System.Web.Services.Description.Import** object.

21 **Insert**

23 [C#] public void Insert(int index, Import import);

24 [C++] public: void Insert(int index, Import\* import);

25 [VB] Public Sub Insert(ByVal index As Integer, ByVal import As Import)

1 [JScript] public function Insert(index : int, import : Import);

3 *Description*

4 Adds the specified **System.Web.Services.Description.Import** instance to  
5 the **ImportCollection** at the specified index.

6 If the number of items in the collection already equals the collection's  
7 capacity, the capacity is doubled by automatically reallocating the internal array  
8 before the new element is inserted. The zero-based index at which to insert *import*.  
9 The **System.Web.Services.Description.Import** to be added to the collection.

10 **Remove**

12 [C#] public void Remove(Import import);

13 [C++] public: void Remove(Import\* import);

14 [VB] Public Sub Remove(ByVal import As Import)

15 [JScript] public function Remove(import : Import);

17 *Description*

18 Removes the first occurrence of the specified  
19 **System.Web.Services.Description.Import** from the **ImportCollection** .

20 This method performs a linear search; therefore, the average execution time  
21 is proportional to **System.Web.Services.Description.ImportCollection.Count** .

22 An **System.Web.Services.Description.Import** object.

23 **SetParent**

25 [C#] protected override void SetParent(object value, object parent);

1 [C++] protected: void SetParent(Object\* value, Object\* parent);

2 [VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As  
3 Object)

4 [JScript] protected override function SetParent(value : Object, parent : Object);

5  
6 *Description*

7 Sets the parent **System.Web.Services.Description.ServiceDescription** of  
8 the specified **System.Web.Services.Description.Import** . An object, of type  
9 **System.Web.Services.Description.Import**, within the collection. The object, of  
10 type **System.Web.Services.Description.ServiceDescription**, to set as the parent.

11 InputBinding class (System.Web.Services.Description)

12 ToString

13  
14  
15 *Description*

16 Provides a set of specifications for data formats and protocols used by the  
17 Web Service for input messages. This class cannot be inherited.

18 InputBinding

19 *Example Syntax:*

20 ToString

21  
22 [C#] public InputBinding();

23 [C++] public: InputBinding();

24 [VB] Public Sub New()

25 [JScript] public function InputBinding();

1 Documentation

2 Extensions

3 ToString

4

5

6 *Description*

7 Gets the

8 **System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect**  
9 **ion** associated with the **InputBinding** instance.

10 Name

11 OperationBinding

12 Message class (System.Web.Services.Description)

13 ToString

14

15

16 *Description*

17 Defines in the abstract the content, either document-oriented or procedure-  
18 oriented, of data passed by the Web Service. This class cannot be inherited.

19 Message

20 *Example Syntax:*

21 ToString

22

23 [C#] public Message();

24 [C++] public: Message();

25

```

1 [VB] Public Sub New()
2 [JScript] public function Message();
3     Documentation
4     Name
5     ToString
6
7
8 Description
9     Gets or sets the name of the Message instance.
10
11     Parts
12     ToString
13
14 [C#] public MessagePartCollection Parts {get;}
15 [C++] public: __property MessagePartCollection* get_Parts();
16 [VB] Public ReadOnly Property Parts As MessagePartCollection
17 [JScript] public function get Parts() : MessagePartCollection;
18
19
20 Description
21     Gets the collection of all the
22 System.Web.Services.Description.MessagePart objects contained in the
23 Message instance.
24
25     ServiceDescription
26     ToString
27
28 [C#] public ServiceDescription ServiceDescription {get;}

```

1 [C++] public: \_\_property ServiceDescription\* get\_ServiceDescription();

2 [VB] Public ReadOnly Property ServiceDescription As ServiceDescription

3 [JScript] public function get ServiceDescription() : ServiceDescription;

4  
5 *Description*

6 Gets the **System.Web.Services.Description.ServiceDescription** of which  
7 the **Message** instance is a member.

8 FindPartByName

9  
10 [C#] public MessagePart FindPartByName(string partName);

11 [C++] public: MessagePart\* FindPartByName(String\* partName);

12 [VB] Public Function FindPartByName(ByVal partName As String) As

13 MessagePart

14 [JScript] public function FindPartByName(partName : String) : MessagePart;

15  
16 *Description*

17 Searches the **System.Web.Services.Description.MessagePartCollection**  
18 returned by the **System.Web.Services.Description.Message.Parts** property, and  
19 returns the named **System.Web.Services.Description.MessagePart** . The string  
20 naming the **System.Web.Services.Description.MessagePart** to be returned.

21 FindPartsByName

22  
23 [C#] public MessagePart[] FindPartsByName(string[] partNames);

24 [C++] public: MessagePart\* FindPartsByName(String\* partNames \_\_gc[]) [];

25 [VB] Public Function FindPartsByName(ByVal partNames() As String) As

1 MessagePart()

2 [JScript] public function FindPartsByName(partNames : String[]) : MessagePart[];

3  
4 *Description*

5 Searches the **System.Web.Services.Description.MessagePartCollection**  
6 returned by the **System.Web.Services.Description.Message.Parts** property and  
7 returns an array of type **System.Web.Services.Description.MessagePart**  
8 containing the named instances.

9 *Return Value:* An array of type **System.Web.Services.Description.MessagePart** .  
10 An array of names of the **System.Web.Services.Description.MessagePart**  
11 instances to be returned.

12 MessageBinding class (System.Web.Services.Description)

13 ToString

14  
15  
16 *Description*

17 Describes how abstract content is mapped into a concrete format. This  
18 abstract class forms the base class for the following classes:

19 **System.Web.Services.Description.FaultBindingSystem.Web.Services.Descript**  
20 **ion.InputBindingSystem.Web.Services.Description.OutputBinding**

21 MessageBinding

22 *Example Syntax:*

23 ToString

24  
25 [C#] protected MessageBinding();

1 [C++] protected: MessageBinding();

2 [VB] Protected Sub New()

3 [JScript] protected function MessageBinding();

4       Documentation

5       Extensions

6       ToString

7

8

9 *Description*

10       Gets the

11 **System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect**

12 **ion** associated with the **MessageBinding** instance.

13       This property is intended to be overridden to provide the format extensions

14 for Input, Output and Fault messages.

15       Name

16       ToString

17

18 [C#] public string Name {get; set;}

19 [C++] public: \_\_property String\* get\_Name();public: \_\_property void

20 set\_Name(String\*);

21 [VB] Public Property Name As String

22 [JScript] public function get Name() : String;public function set Name(String);

23

24 *Description*

25



1 Gets or sets the name of the

2 **System.Web.Services.Description.MessageBinding** .

3 OperationBinding

4 ToString

5  
6 [C#] public OperationBinding OperationBinding {get;}

7 [C++] public: \_\_property OperationBinding\* get\_OperationBinding();

8 [VB] Public ReadOnly Property OperationBinding As OperationBinding

9 [JScript] public function get OperationBinding() : OperationBinding;

10  
11 *Description*

12 Gets the **System.Web.Services.Description.OperationBinding** of which  
13 the **MessageBinding** instance is a member.

14 MessageCollection class (System.Web.Services.Description)

15 ToString

16  
17  
18 *Description*

19 Represents a collection of **System.Web.Services.Description.Message**  
20 objects. This class cannot be inherited.

21 Count

22 InnerList

23 Item

24 ToString

## System.Web.Services.Description.Message

### *Description*

Gets or sets the value of a **System.Web.Services.Description.Message** at the specified zero-based index. The zero-based index of the **Message** whose value is modified or returned.

Item

ToString

[C#] public Message this[string name] {get;}

[C++] public: \_\_property Message\* get\_Item(String\* name);

[VB] Public Default ReadOnly Property Item(ByVal name As String) As Message

[JScript] returnValue = MessageCollectionObject.Item(name);

### *Description*

Gets the **Message** by its **System.Web.Services.Description.Message.Name** property. The string value representing the name of the **Message** returned.

List

Table

Add

[C#] public int Add(Message message);

[C++] public: int Add(Message\* message);

[VB] Public Function Add(ByVal message As Message) As Integer

1 [JScript] public function Add(message : Message) : int;

2  
3 *Description*

4 Adds the specified **System.Web.Services.Description.Message** instance to  
5 the end of the **MessageCollection** .

6 *Return Value:* Returns the zero-based index where *message* has been added. The  
7 **System.Web.Services.Description.Message** to be added to the  
8 **MessageCollection** .

9 Contains

10  
11 [C#] public bool Contains(Message message);

12 [C++] public: bool Contains(Message\* message);

13 [VB] Public Function Contains(ByVal message As Message) As Boolean

14 [JScript] public function Contains(message : Message) : Boolean;

15  
16 *Description*

17 Gets a value indicating whether the specified  
18 **System.Web.Services.Description.Message** instance is a member of the  
19 **MessageCollection** .

20 *Return Value:* **true** if *message* is a member of the **MessageCollection** ; otherwise,  
21 **false** . A **System.Web.Services.Description.Message** object.

22 CopyTo

23  
24 [C#] public void CopyTo(Message[] array, int index);

25 [C++] public: void CopyTo(Message\* array[], int index);

1 [VB] Public Sub CopyTo(ByVal array() As Message, ByVal index As Integer)  
2 [JScript] public function CopyTo(array : Message[], index : int);

3  
4 *Description*

5 Copies the entire **MessageCollection** to a compatible one-dimensional  
6 array of type **System.Web.Services.Description.Message** , starting at the  
7 specified zero-based index of the target array. The array, of type  
8 **System.Web.Services.Description.Message**, serving as the destination of the  
9 copy action. The zero-based index at which to start placing the copied collection.

10 **GetKey**

11  
12 [C#] protected override string GetKey(object value);  
13 [C++] protected: String\* GetKey(Object\* value);  
14 [VB] Overrides Protected Function GetKey(ByVal value As Object) As String  
15 [JScript] protected override function GetKey(value : Object) : String;

16  
17 *Description*

18 Supplies the name of the **System.Web.Services.Description.Message**  
19 associated with the value passed by reference. A  
20 **System.Web.Services.Description.Message** instance for which to return the  
21 name.

22 **IndexOf**

23  
24 [C#] public int IndexOf(Message message);  
25 [C++] public: int IndexOf(Message\* message);

1 [VB] Public Function IndexOf(ByVal message As Message) As Integer

2 [JScript] public function IndexOf(message : Message) : int;

3  
4 *Description*

5 Searches for the specified **System.Web.Services.Description.Message**  
6 instance and returns the zero-based index of the first occurrence within the  
7 collection.

8 *Return Value:* Returns a 32-bit signed integer. A

9 **System.Web.Services.Description.Message** object for which to search the  
10 **MessageCollection**.

11 **Insert**

12  
13 [C#] public void Insert(int index, Message message);

14 [C++] public: void Insert(int index, Message\* message);

15 [VB] Public Sub Insert(ByVal index As Integer, ByVal message As Message)

16 [JScript] public function Insert(index : int, message : Message);

17  
18 *Description*

19 Adds the specified **System.Web.Services.Description.Message** instance to  
20 the **MessageCollection** at the specified index.

21 If the number of items in the collection already equals the collection's  
22 capacity, the capacity is doubled by automatically reallocating the internal array  
23 before the new element is inserted. The zero-based index at which to insert  
24 *message*. The **System.Web.Services.Description.Message** instance to add to the  
25 collection.

## Remove

```
[C#] public void Remove(Message message);  
[C++] public: void Remove(Message* message);  
[VB] Public Sub Remove(ByVal message As Message)  
[JScript] public function Remove(message : Message);
```

### *Description*

Removes the first occurrence of the specified **System.Web.Services.Description.Message** instance from the **MessageCollection**.

This method performs a linear search; therefore, the average execution time is proportional to **System.Web.Services.Description.MessageCollection.Count**. A **System.Web.Services.Description.Message** instance for which to search the collection.

## SetParent

```
[C#] protected override void SetParent(object value, object parent);  
[C++] protected: void SetParent(Object* value, Object* parent);  
[VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As  
Object)  
[JScript] protected override function SetParent(value : Object, parent : Object);
```

### *Description*

1       Sets the parent **System.Web.Services.Description.ServiceDescription** of  
2 a member of the **MessageCollection** . An object, of type  
3 **System.Web.Services.Description.Message**, within the collection. The object, of  
4 type **System.Web.Services.Description.ServiceDescription**, to be set as the  
5 parent.

6       MessagePart class (System.Web.Services.Description)

7       ToString

8  
9  
10    *Description*

11       Allows messages to be broken up into their logical units, with specific  
12 abstract information for each part. This class cannot be inherited.

13       MessagePart

14       *Example Syntax:*

15       ToString

16  
17    [C#] public MessagePart();

18    [C++] public: MessagePart();

19    [VB] Public Sub New()

20    [JScript] public function MessagePart();

21       Documentation

22       Element

23       ToString

1  
2  
3 *Description*

4 Gets or sets the XML element name of the  
5 **System.Web.Services.Description.MessagePart** instance.

6 This property corresponds to the attribute of the tag for which the  
7 MessagePart class serves as a wrapper.

8 Message

9 ToString

10  
11 [C#] public Message Message {get;}

12 [C++] public: \_\_property Message\* get\_Message();

13 [VB] Public ReadOnly Property Message As Message

14 [JScript] public function get Message() : Message;

15  
16 *Description*

17 Gets the **System.Web.Services.Description.Message** of which the  
18 **MessagePart** instance is a member.

19 Name

20 ToString

21  
22 [C#] public string Name {get; set;}

23 [C++] public: \_\_property String\* get\_Name();public: \_\_property void  
24 set\_Name(String\*);

25 [VB] Public Property Name As String



1 [JScript] public function get Name() : String;public function set Name(String);

2  
3 *Description*

4 Gets or sets the name of the  
5 **System.Web.Services.Description.MessagePart** instance.

6 Returns an empty string ("" ) if the property value has not been assigned.  
7 This property corresponds to the attribute of the tag for which the **MessagePart**  
8 class serves as a wrapper.

9 Type

10 ToString

11  
12 [C#] public XmlQualifiedName Type {get; set;}

13 [C++] public: \_\_property XmlQualifiedName\* get\_Type();public: \_\_property  
14 void set\_Type(XmlQualifiedName\*);

15 [VB] Public Property Type As XmlQualifiedName

16 [JScript] public function get Type() : XmlQualifiedName;public function set  
17 Type(XmlQualifiedName);

18  
19 *Description*

20 Gets or sets the XML datatype of the  
21 **System.Web.Services.Description.MessagePart** instance.

22 This property corresponds to the attribute of the tag for which the  
23 **MessagePart** class serves as a wrapper. In general it refers to an XSD datatype,  
24 but can also be extended as long as the XML namespace used is different from  
25 that of WSDL.

1 MessagePartCollection class (System.Web.Services.Description)

2 ToString

3  
4  
5 *Description*

6 Represents a collection of

7 **System.Web.Services.Description.MessagePart** objects. This class cannot be  
8 inherited.

9 Count

10 InnerList

11 Item

12 ToString

13 **System.Web.Services.Description.MessagePart**

14  
15 *Description*

16 Gets or sets the value of a

17 **System.Web.Services.Description.MessagePart** at the specified zero-based  
18 index. The zero-based index of the **MessagePart** whose value is modified or  
19 returned.

20 Item

21 ToString

22  
23 [C#] public MessagePart this[string name] {get;}

24 [C++] public: \_\_property MessagePart\* get\_Item(String\* name);

25 [VB] Public Default ReadOnly Property Item(ByVal name As String) As

## MessagePart

[JScript] returnValue = MessagePartCollectionObject.Item(name);

### *Description*

Gets the **MessagePart** by its **Name** property. The string value representing the name of the **System.Web.Services.Description.MessagePart** returned.

List

Table

Add

[C#] public int Add(MessagePart messagePart);

[C++] public: int Add(MessagePart\* messagePart);

[VB] Public Function Add(ByVal messagePart As MessagePart) As Integer

[JScript] public function Add(messagePart : MessagePart) : int;

### *Description*

Adds the specified **System.Web.Services.Description.MessagePart** to the end of the **System.Web.Services.Description.MessagePartCollection**.

*Return Value:* Returns the index where *messagePart* has been added. The **System.Web.Services.Description.MessagePart** to be added to the collection.

Contains

[C#] public bool Contains(MessagePart messagePart);

[C++] public: bool Contains(MessagePart\* messagePart);

[VB] Public Function Contains(ByVal messagePart As MessagePart) As Boolean

1 [JScript] public function Contains(messagePart : MessagePart) : Boolean;

3 *Description*

4 Gets a value indicating whether the specified  
5 **System.Web.Services.Description.MessagePart** is a member of the  
6 **MessagePartCollection** .

7 *Return Value:* **true** if *messagePart* is a member of the **MessagePartCollection** ;  
8 otherwise, **false** . A **System.Web.Services.Description.MessagePart** object.

9 CopyTo

11 [C#] public void CopyTo(MessagePart[] array, int index);

12 [C++] public: void CopyTo(MessagePart\* array[], int index);

13 [VB] Public Sub CopyTo(ByVal array() As MessagePart, ByVal index As Integer)

14 [JScript] public function CopyTo(array : MessagePart[], index : int);

16 *Description*

17 Copies the entire **MessagePartCollection** to a compatible one-dimensional  
18 array of type **System.Web.Services.Description.MessagePart** , starting at the  
19 specified zero-based index of the target array. An array of type  
20 **System.Web.Services.Description.MessagePart** serving as the destination of the  
21 copy action. The zero-based index at which to start placing the copied collection.

22 GetKey

24 [C#] protected override string GetKey(object value);

25 [C++] protected: String\* GetKey(Object\* value);

1 [VB] Overrides Protected Function GetKey(ByVal value As Object) As String

2 [JScript] protected override function GetKey(value : Object) : String;

3  
4 *Description*

5 Returns the name of the **System.Web.Services.Description.MessagePart**  
6 associated with the value passed by reference. An object for which to return the  
7 name.

8 IndexOf

9  
10 [C#] public int IndexOf(MessagePart messagePart);

11 [C++] public: int IndexOf(MessagePart\* messagePart);

12 [VB] Public Function IndexOf(ByVal messagePart As MessagePart) As Integer

13 [JScript] public function IndexOf(messagePart : MessagePart) : int;

14  
15 *Description*

16 Searches for the specified  
17 **System.Web.Services.Description.MessagePart** and returns the zero-based  
18 index of the first occurrence within the collection.

19 *Return Value:* Returns a 32-bit signed integer. A

20 **System.Web.Services.Description.MessagePart** object.

21 Insert

22  
23 [C#] public void Insert(int index, MessagePart messagePart);

24 [C++] public: void Insert(int index, MessagePart\* messagePart);

25 [VB] Public Sub Insert(ByVal index As Integer, ByVal messagePart As

1 MessagePart)

2 [JScript] public function Insert(index : int, messagePart : MessagePart);

3  
4 *Description*

5 Adds the specified **System.Web.Services.Description.MessagePart** to the  
6 **MessagePartCollection** at the specified index.

7 If the number of items in the collection already equals the collection's  
8 capacity, the capacity is doubled by automatically reallocating the internal array  
9 before the new element is inserted. The zero-based index at which to insert  
10 *messagePart*. The **System.Web.Services.Description.MessagePart** to be added  
11 to the collection.

12 Remove

13  
14 [C#] public void Remove(MessagePart messagePart);

15 [C++] public: void Remove(MessagePart\* messagePart);

16 [VB] Public Sub Remove(ByVal messagePart As MessagePart)

17 [JScript] public function Remove(messagePart : MessagePart);

18  
19 *Description*

20 Removes the first occurrence of the specified  
21 **System.Web.Services.Description.MessagePart** from the  
22 **MessagePartCollection** .

23 This method performs a linear search; therefore, the average execution time  
24 is proportional to

1 **System.Web.Services.Description.MessagePartCollection.Count** . A

2 **System.Web.Services.Description.MessagePart** object.

3     SetParent

5 [C#] protected override void SetParent(object value, object parent);

6 [C++] protected: void SetParent(Object\* value, Object\* parent);

7 [VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As  
8 Object)

9 [JScript] protected override function SetParent(value : Object, parent : Object);

11 *Description*

12         Sets the parent **System.Web.Services.Description.Message** of a member  
13 of **System.Web.Services.Description.MessagePartCollection** . An object, of  
14 type **System.Web.Services.Description.MessagePart**, within the collection. The  
15 object, of type **System.Web.Services.Description.Message**, to set as the parent.

16         MimeContentBinding class (System.Web.Services.Description)

17         ToString

20 *Description*

21         Represents an extensibility element added to an  
22 **System.Web.Services.Description.InputBinding** or an  
23 **System.Web.Services.Description.OutputBinding** within a Web Service,  
24 specifying the MIME format for the body of the HTTP transmission. This class  
25 cannot be inherited.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

ToString

```
[C#] public const string Namespace;  
[C++] public: const String* Namespace;  
[VB] Public Const Namespace As String  
[JScript] public var Namespace : String;
```

*Description*

Specifies the URI for the XML namespace of the **MimeContentBinding** class. This field is constant.

MimeContentBinding

*Example Syntax:*

ToString

```
[C#] public MimeContentBinding();  
[C++] public: MimeContentBinding();  
[VB] Public Sub New()  
[JScript] public function MimeContentBinding();
```

- Handled
- Parent
- Part
- ToString

*Description*



1 Gets or sets the name of the  
2 **System.Web.Services.Description.MessagePart** to which the  
3 **MimeContentBinding** applies.

4 Required

5 Type

6 ToString

7  
8  
9 *Description*

10 Gets or sets a value indicating the format of the body of the HTTP  
11 transmission.

12 The WSDL specification defines bindings for several MIME types,  
13 including text/xml, multipart/related, and application/x-www-form-urlencoded,  
14 although any MIME type may be used. The wildcard character (\*) may also be  
15 used. For example, the string "text/\*" would represent all text types. If the value of  
16 this property is not set, the **MimeContentBinding** specifies all MIME types.

17 MimeMultipartRelatedBinding class (System.Web.Services.Description)

18 ToString

19  
20  
21 *Description*

22 Represents an extensibility element added to an  
23 **System.Web.Services.Description.InputBinding** or an  
24 **System.Web.Services.Description.OutputBinding** , specifying the individual  
25

1 MIME formats for each **System.Web.Services.Description.MessagePart** within  
2 the HTTP transmission. This class cannot be inherited.

3 **MimeMultipartRelatedBinding**

4 *Example Syntax:*

5 **ToString**

6  
7 [C#] public MimeMultipartRelatedBinding();

8 [C++] public: MimeMultipartRelatedBinding();

9 [VB] Public Sub New()

10 [JScript] public function MimeMultipartRelatedBinding();

11 **Handled**

12 **Parent**

13 **Parts**

14 **ToString**

15  
16  
17 *Description*

18 Gets the collection of extensibility elements added to the  
19 **MimeMultipartRelatedBinding** , which specify the MIME format for the  
20 corresponding **System.Web.Services.Description.MessagePart** instances.

21 **Required**

22 **MimePart** class (System.Web.Services.Description)

23 **ToString**

### *Description*

Represents an extensibility element added to a **System.Web.Services.Description.MimeMultipartRelatedBinding** , specifying the concrete MIME type for the **System.Web.Services.Description.MessagePart** to which the **MimePart** applies. This class cannot be inherited.

MimePart

### *Example Syntax:*

ToString

[C#] public MimePart();

[C++] public: MimePart();

[VB] Public Sub New()

[JScript] public function MimePart();

Extensions

ToString

[C#] public ServiceDescriptionFormatExtensionCollection Extensions {get;}

[C++] public: \_\_property ServiceDescriptionFormatExtensionCollection\*

get\_Extensions();

[VB] Public ReadOnly Property Extensions As

ServiceDescriptionFormatExtensionCollection

[JScript] public function get Extensions() :

ServiceDescriptionFormatExtensionCollection;

1  
2 *Description*

3 Gets the collection of bindings for the  
4 **System.Web.Services.Description.MimeMultipartRelatedBinding** of which the  
5 **MimePart** is a member.

6 This collection must have at least one member. If it has more than one  
7 member, then the members are alternative MIME formats for a corresponding  
8 **System.Web.Services.Description.MessagePart** .

9 Handled

10 Parent

11 Required

12 MimePartCollection class (System.Web.Services.Description)

13 ToString

14  
15  
16 *Description*

17 Represents a collection of **System.Web.Services.Description.MimePart**  
18 objects. This class cannot be inherited.

19 MimePartCollection

20 *Example Syntax:*

21 ToString

22  
23 [C#] public MimePartCollection();

24 [C++] public: MimePartCollection();

1 [VB] Public Sub New()

2 [JScript] public function MimePartCollection();

3 Count

4 InnerList

5 Item

6 ToString

7  
8  
9 *Description*

10 Gets or sets the value of a **System.Web.Services.Description.MimePart** at  
11 the specified zero-based index. The zero-based index of the **MimePart** whose  
12 value is modified or returned.

13 List

14 Add

15  
16 [C#] public int Add(MimePart mimePart);

17 [C++] public: int Add(MimePart\* mimePart);

18 [VB] Public Function Add(ByVal mimePart As MimePart) As Integer

19 [JScript] public function Add(mimePart : MimePart) : int;

20  
21 *Description*

22 Adds the specified **System.Web.Services.Description.MimePart** instance  
23 to the end of the **MimePartCollection** .

24 *Return Value:* Returns the zero-based index where *mimePart* has been added. The  
25 **System.Web.Services.Description.MimePart** to be added to the collection.

## Contains

[C#] public bool Contains(MimePart mimePart);

[C++] public: bool Contains(MimePart\* mimePart);

[VB] Public Function Contains(ByVal mimePart As MimePart) As Boolean

[JScript] public function Contains(mimePart : MimePart) : Boolean;

### *Description*

Gets a value indicating whether the specified

**System.Web.Services.Description.MimePart** instance is a member of the **MimePartCollection**.

*Return Value:* **true** if *mimePart* is a member of the **MimePartCollection**; otherwise, **false**. A **System.Web.Services.Description.MimePart** instance.

### CopyTo

[C#] public void CopyTo(MimePart[] array, int index);

[C++] public: void CopyTo(MimePart\* array[], int index);

[VB] Public Sub CopyTo(ByVal array() As MimePart, ByVal index As Integer)

[JScript] public function CopyTo(array : MimePart[], index : int);

### *Description*

Copies the entire **MimePartCollection** to a compatible one-dimensional array of type **System.Web.Services.Description.MimePart**, starting at the specified zero-based index of the target array. An array of type

1 **System.Web.Services.Description.MimePart** serving as the destination of the  
2 copy action. The zero-based index at which to start placing the copied collection.

3 **IndexOf**

4  
5 [C#] public int IndexOf(MimePart mimePart);

6 [C++] public: int IndexOf(MimePart\* mimePart);

7 [VB] Public Function IndexOf(ByVal mimePart As MimePart) As Integer

8 [JScript] public function IndexOf(mimePart : MimePart) : int;

9  
10 *Description*

11 Searches for the specified **System.Web.Services.Description.MimePart**  
12 instance and returns the zero-based index of the first occurrence within the  
13 collection.

14 *Return Value:* Returns a 32-bit signed integer. A

15 **System.Web.Services.Description.MimePart** object for which to search the  
16 **MimePartCollection**.

17 **Insert**

18  
19 [C#] public void Insert(int index, MimePart mimePart);

20 [C++] public: void Insert(int index, MimePart\* mimePart);

21 [VB] Public Sub Insert(ByVal index As Integer, ByVal mimePart As MimePart)

22 [JScript] public function Insert(index : int, mimePart : MimePart);

23  
24 *Description*

25

1 Adds the specified **System.Web.Services.Description.MimePart** instance  
2 to the **MimePartCollection** at the specified index.

3 If the number of items in the collection already equals the collection's  
4 capacity, the capacity is doubled by automatically reallocating the internal array  
5 before the new element is inserted. The zero-based index at which to insert  
6 *mimePart*. The **System.Web.Services.Description.MimePart** to add to the  
7 collection.

8 Remove

9  
10 [C#] public void Remove(MimePart mimePart);

11 [C++] public: void Remove(MimePart\* mimePart);

12 [VB] Public Sub Remove(ByVal mimePart As MimePart)

13 [JScript] public function Remove(mimePart : MimePart);

14  
15 *Description*

16 Removes the first occurrence of the specified  
17 **System.Web.Services.Description.MimePart** from the **MimePartCollection** .

18 This method performs a linear search; therefore, the average execution time  
19 is proportional to

20 **System.Web.Services.Description.MimePartCollection.Count** . A

21 **System.Web.Services.Description.MimePart** instance for which to search the  
22 collection.

23 MimeTextBinding class (System.Web.Services.Description)

24 ToString



### *Description*

Represents an extensibility element added to an **System.Web.Services.Description.InputBinding** , an **System.Web.Services.Description.OutputBinding** or a **System.Web.Services.Description.MimePart** , specifying the text patterns for which to search the HTTP transmission. This class cannot be inherited.

### *ToString*

```
[C#] public const string Namespace;  
[C++] public: const String* Namespace;  
[VB] Public Const Namespace As String  
[JScript] public var Namespace : String;
```

### *Description*

Specifies the URI for the XML namespace of the **MimeTextBinding** class.  
This field is constant.

### *MimeTextBinding*

### *Example Syntax:*

### *ToString*

```
[C#] public MimeTextBinding();  
[C++] public: MimeTextBinding();
```

1 [VB] Public Sub New()

2 [JScript] public function MimeTextBinding();

3     Handled

4     Matches

5     ToString

6  
7  
8 *Description*

9     Gets the collection of MIME text patterns for which the HTTP transmission  
10 is searched.

11     Parent

12     Required

13     MimeTextMatch class (System.Web.Services.Description)

14     ToString

15  
16  
17 *Description*

18     Represents a text pattern for which the HTTP transmission is searched. This  
19 class cannot be inherited.

20     MimeTextMatch

21     *Example Syntax:*

22     ToString

23  
24 [C#] public MimeTextMatch();

25 [C++] public: MimeTextMatch();

```

1 [VB] Public Sub New()
2 [JScript] public function MimeTextMatch();
3     Capture
4     ToString
5
6 [C#] public int Capture {get; set;}
7 [C++] public: __property int get_Capture();public: __property void
8 set_Capture(int);
9 [VB] Public Property Capture As Integer
10 [JScript] public function get Capture() : int;public function set Capture(int);
11

```

### *Description*

Returns a 32-bit signed integer. The default value is 0.

### *Group*

### *ToString*

```

17 [C#] public int Group {get; set;}
18 [C++] public: __property int get_Group();public: __property void set_Group(int);
19 [VB] Public Property Group As Integer
20 [JScript] public function get Group() : int;public function set Group(int);
21

```

### *Description*

Gets or sets a value indicating the number of groups in which to place the results of the text search.

The value of this property should be less than or equal to the value of the **System.Web.Services.Description.MimeTextMatch.Repeats** property. It is recommended that developers use the default value.

IgnoreCase

ToString

[C#] public bool IgnoreCase {get; set;}

[C++] public: \_\_property bool get\_IgnoreCase();public: \_\_property void set\_IgnoreCase(bool);

[VB] Public Property IgnoreCase As Boolean

[JScript] public function get IgnoreCase() : Boolean;public function set IgnoreCase(Boolean);

#### *Description*

Gets or sets a value indicating whether the search should disregard the case of the text to be searched.

Matches

ToString

[C#] public MimeTextMatchCollection Matches {get;}

[C++] public: \_\_property MimeTextMatchCollection\* get\_Matches();

[VB] Public ReadOnly Property Matches As MimeTextMatchCollection

[JScript] public function get Matches() : MimeTextMatchCollection;

#### *Description*

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

Name

ToString

```
[C#] public string Name {get; set;}
[C++] public: __property String* get_Name();public: __property void
set_Name(String*);
[VB] Public Property Name As String
[JScript] public function get Name() : String;public function set Name(String);
```

*Description*

Gets or sets the name of the **MimeTextMatch** instance.

Pattern

ToString

```
[C#] public string Pattern {get; set;}
[C++] public: __property String* get_Pattern();public: __property void
set_Pattern(String*);
[VB] Public Property Pattern As String
[JScript] public function get Pattern() : String;public function set Pattern(String);
```

*Description*

Gets or sets the text pattern for the search.

The value of this property can contain wildcard characters.

Repeats

ToString

[C#] public int Repeats {get; set;}

[C++] public: \_\_property int get\_Repeats();public: \_\_property void  
set\_Repeats(int);

[VB] Public Property Repeats As Integer

[JScript] public function get Repeats() : int;public function set Repeats(int);

*Description*

Gets or sets a value indicating the number of times the search is to be performed.

In a case where there is more than one possible match, a property value of 1 will return only the first match found.

RepeatsString

ToString

[C#] public string RepeatsString {get; set;}

[C++] public: \_\_property String\* get\_RepeatsString();public: \_\_property void  
set\_RepeatsString(String\*);

[VB] Public Property RepeatsString As String

[JScript] public function get RepeatsString() : String;public function set  
RepeatsString(String);

*Description*

1 Gets or sets a value indicating the number of times the search is to be  
2 performed.

3 This property returns the same information as the  
4 **System.Web.Services.Description.MimeTextMatch.Repeats** property, but as a  
5 string rather than a 32-bit signed integer. A value of "\*" corresponds to  
6 **System.Int32.MaxValue** .

7 Type

8 ToString

9  
10 [C#] public string Type {get; set;}

11 [C++] public: \_\_property String\* get\_Type();public: \_\_property void  
12 set\_Type(String\*);

13 [VB] Public Property Type As String

14 [JScript] public function get Type() : String;public function set Type(String);

15  
16 *Description*

17 Gets or sets a value indicating the MIME format of the text to be searched.

18 This property returns an empty string ("") if its value has not been set.

19 MimeTextMatchCollection class (System.Web.Services.Description)

20 ToString

21  
22  
23 *Description*

Provides a collection of  
**System.Web.Services.Description.MimeTextMatch** instances. This class cannot  
be inherited.

MimeTextMatchCollection

*Example Syntax:*

ToString

[C#] public MimeTextMatchCollection();

[C++] public: MimeTextMatchCollection();

[VB] Public Sub New()

[JScript] public function MimeTextMatchCollection();

Count

InnerList

Item

ToString

### *Description*

Gets or sets the value of a member of the **MimeTextMatchCollection** at  
the specified zero-based index. The zero-based index of the **MimeTextMatch**  
whose value is returned or modified.

List

Add

[C#] public int Add(MimeTextMatch match);



1 [C++] public: int Add(MimeTextMatch\* match);

2 [VB] Public Function Add(ByVal match As MimeTextMatch) As Integer

3 [JScript] public function Add(match : MimeTextMatch) : int;

4  
5 *Description*

6 Adds the specified **System.Web.Services.Description.MimeTextMatch**  
7 instance to the end of the **MimeTextMatchCollection** .

8 *Return Value:* Returns the zero-based index where *match* has been added. The  
9 **System.Web.Services.Description.MimeTextMatch** to be added to the  
10 collection.

11 Contains

12  
13 [C#] public bool Contains(MimeTextMatch match);

14 [C++] public: bool Contains(MimeTextMatch\* match);

15 [VB] Public Function Contains(ByVal match As MimeTextMatch) As Boolean

16 [JScript] public function Contains(match : MimeTextMatch) : Boolean;

17  
18 *Description*

19 Gets a value indicating whether the specified  
20 **System.Web.Services.Description.MimeTextMatch** instance is a member of the  
21 **MimeTextMatchCollection** .

22 *Return Value:* **true** if *match* is a member of the **MimeTextMatchCollection** ;  
23 otherwise, **false** . A **System.Web.Services.Description.MimeTextMatch**  
24 instance.

25 CopyTo

```

1 [C#] public void CopyTo(MimeTextMatch[] array, int index);
2
3 [C++] public: void CopyTo(MimeTextMatch* array[], int index);
4
5 [VB] Public Sub CopyTo(ByVal array() As MimeTextMatch, ByVal index As
6 Integer)
7
8 [JScript] public function CopyTo(array : MimeTextMatch[], index : int);

```

### *Description*

Copies the entire **MimeTextMatchCollection** to a compatible one-dimensional array of type **System.Web.Services.Description.MimeTextMatch** , starting at the specified zero-based index of the target array. An array of type **System.Web.Services.Description.MimeTextMatch** serving as the destination of the copy action. The zero-based index at which to start placing the copied collection.

### *IndexOf*

```

17 [C#] public int IndexOf(MimeTextMatch match);
18
19 [C++] public: int IndexOf(MimeTextMatch* match);
20
21 [VB] Public Function IndexOf(ByVal match As MimeTextMatch) As Integer
22
23 [JScript] public function IndexOf(match : MimeTextMatch) : int;

```

### *Description*

Searches for the specified **System.Web.Services.Description.MimeTextMatch** instance and returns the zero-based index of the first occurrence within the collection.

1 *Return Value:* Returns a 32-bit signed integer. A

2 **System.Web.Services.Description.MimeTextMatch** object for which to search  
3 the **MimeTextMatchCollection**.

#### 4 Insert

5  
6 [C#] public void Insert(int index, MimeTextMatch match);

7 [C++] public: void Insert(int index, MimeTextMatch\* match);

8 [VB] Public Sub Insert(ByVal index As Integer, ByVal match As  
9 MimeTextMatch)

10 [JScript] public function Insert(index : int, match : MimeTextMatch);

#### 11 12 *Description*

13 Adds the specified **System.Web.Services.Description.MimeTextMatch**  
14 instance to the **MimeTextMatchCollection** at the specified index.

15 If the number of items in the collection already equals the collection's  
16 capacity, the capacity is doubled by automatically reallocating the internal array  
17 before the new element is inserted. The zero-based index at which to insert *match*.  
18 The **System.Web.Services.Description.MimeTextMatch** to add to the collection.

#### 19 Remove

20  
21 [C#] public void Remove(MimeTextMatch match);

22 [C++] public: void Remove(MimeTextMatch\* match);

23 [VB] Public Sub Remove(ByVal match As MimeTextMatch)

24 [JScript] public function Remove(match : MimeTextMatch);

25

1  
2 *Description*

3 Removes the first occurrence of the specified  
4 **System.Web.Services.Description.MimeTextMatch** instance from the  
5 **MimeTextMatchCollection** .

6 This method performs a linear search; therefore, the average execution time  
7 is proportional to

8 **System.Web.Services.Description.MimeTextMatchCollection.Count** . A  
9 **System.Web.Services.Description.MimeTextMatch** instance for which to search  
10 the collection.

11 MimeXmlBinding class (System.Web.Services.Description)

12 ToString

13  
14  
15 *Description*

16 Represents an extensibility element added to a  
17 **System.Web.Services.Description.MimePart** , an  
18 **System.Web.Services.Description.InputBinding** or an  
19 **System.Web.Services.Description.OutputBinding** . Specifies the schema for  
20 XML messages that are not SOAP compliant. This class cannot be inherited.

21 MimeXmlBinding

22 *Example Syntax:*

23 ToString

24  
25 [C#] public MimeXmlBinding();

```

1 [C++] public: MimeXmlBinding();
2 [VB] Public Sub New()
3 [JScript] public function MimeXmlBinding();
4     Handled
5     Parent
6     Part
7     ToString
8
9

```

#### 10 *Description*

11 Gets or sets the name of the

12 **System.Web.Services.Description.MessagePart** to which the **MimeXmlBinding**

13 applies.

14 Required

15 Operation class (System.Web.Services.Description)

16 ToString

#### 19 *Description*

20 Provides an abstract definition of an action supported by the Web Service.

21 This class cannot be inherited.

22 Exactly one instance of this class will be a member of the

23 **System.Web.Services.Description.Operation.Messages** property of the parent

24 **System.Web.Services.Description.Operation** .

25 Operation

*Example Syntax:*

ToString

[C#] public Operation();

[C++] public: Operation();

[VB] Public Sub New()

[JScript] public function Operation();

Documentation

Faults

ToString

### *Description*

Gets the collection of faults, or error messages, defined by the **System.Web.Services.Description.Operation** instance.

Messages

ToString

[C#] public OperationMessageCollection Messages {get;}

[C++] public: \_\_property OperationMessageCollection\* get\_Messages();

[VB] Public ReadOnly Property Messages As OperationMessageCollection

[JScript] public function get Messages() : OperationMessageCollection;

### *Description*

1 Gets the collection of **System.Web.Services.Description.Message**  
2 instances defined by the **System.Web.Services.Description.Operation** instance.

3 Because an **Operation** instance is associated with exactly one  
4 **System.Web.Services.Description.OperationInput** and exactly one  
5 **System.Web.Services.Description.OperationOutput** , only one instance of each  
6 can be a member of this collection, and thus the collection can have a maximum of  
7 two members.

8 Name

9 ToString

10  
11 [C#] public string Name {get; set;}

12 [C++] public: \_\_property String\* get\_Name();public: \_\_property void  
13 set\_Name(String\*);

14 [VB] Public Property Name As String

15 [JScript] public function get Name() : String;public function set Name(String);

16  
17 *Description*

18 Gets or sets the name of the **System.Web.Services.Description.Operation**  
19 instance.

20 Returns an empty string ("" ) if the property value has not been set.

21 ParameterOrder

22 ToString

23  
24 [C#] public string[] ParameterOrder {get; set;}

25 [C++] public: \_\_property String\* get\_ParameterOrder();public: \_\_property void

1 set\_ParameterOrder(String\* \_\_gc[]);

2 [VB] Public Property ParameterOrder As String ()

3 [JScript] public function get ParameterOrder() : String[];public function set

4 ParameterOrder(String[]);

5  
6 *Description*

7 Gets or sets an array of elements contained in the

8 **System.Web.Services.Description.Operation.ParameterOrderString** .

9 Note that the elements of this array are identical to those of the

10 **System.Web.Services.Description.Operation.ParameterOrderString** , but

11 housed in an array rather than a space-delimited string.

12 ParameterOrderString

13 ToString

14  
15 [C#] public string ParameterOrderString {get; set;}

16 [C++] public: \_\_property String\* get \_ParameterOrderString();public: \_\_property

17 void set \_ParameterOrderString(String\*);

18 [VB] Public Property ParameterOrderString As String

19 [JScript] public function get ParameterOrderString() : String;public function set

20 ParameterOrderString(String);

21  
22 *Description*

23 An optional RPC-signature ordering specification for request/response or

24 solicit/response operations. The value is a list of names of



1 **System.Web.Services.Description.MessagePart** instances separated by a single  
2 space.

3 The named **System.Web.Services.Description.MessagePart** instances  
4 must adhere to the following rules: The order of the named  
5 **System.Web.Services.Description.MessagePart** instances reflects the order of  
6 the parameters in the RPC signature.

7 PortType

8 ToString

9  
10 [C#] public PortType PortType {get;}

11 [C++] public: \_\_property PortType\* get\_PortType();

12 [VB] Public ReadOnly Property PortType As PortType

13 [JScript] public function get PortType() : PortType;

14  
15 *Description*

16 Gets the **System.Web.Services.Description.PortType** of which the  
17 **Operation** is a member.

18 IsBoundBy

19  
20 [C#] public bool IsBoundBy(OperationBinding operationBinding);

21 [C++] public: bool IsBoundBy(OperationBinding\* operationBinding);

22 [VB] Public Function IsBoundBy(ByVal operationBinding As OperationBinding)

23 As Boolean

24 [JScript] public function IsBoundBy(operationBinding : OperationBinding) :

25 Boolean;

## Description

Gets a value indicating whether the specified **System.Web.Services.Description.OperationBinding** is used in the **Operation** instance.

**Return Value:** **true** if the **Operation** instance uses *operationBinding* ; otherwise, **false** . An **System.Web.Services.Description.OperationBinding** instance to be checked to determine whether it is used by the **Operation**.

OperationBinding class (System.Web.Services.Description)

ToString

## Description

Provides specifications for protocols and data formats for the messages used in the action supported by the Web Service. This class cannot be inherited.

This class contains all the **System.Web.Services.Description.Binding** elements required for the actions supported by the Web Service.

OperationBinding

*Example Syntax:*

ToString

[C#] public OperationBinding();

[C++] public: OperationBinding();

[VB] Public Sub New()

[JScript] public function OperationBinding();

Binding

ToString

[C#] public Binding Binding {get;}

[C++] public: \_\_property Binding\* get\_Binding();

[VB] Public ReadOnly Property Binding As Binding

[JScript] public function get Binding() : Binding;

### *Description*

The **System.Web.Services.Description.Binding** of which the **System.Web.Services.Description.OperationBinding** instance is a member.

Documentation

Extensions

ToString

### *Description*

Gets the

**System.Web.Services.Description.ServiceDescriptionFormatExtensionCollection** specific to the **System.Web.Services.Description.OperationBinding** instance.

Faults

ToString

[C#] public FaultBindingCollection Faults {get;}

```

1 [C++] public: __property FaultBindingCollection* get_Faults();
2 [VB] Public ReadOnly Property Faults As FaultBindingCollection
3 [JScript] public function get Faults() : FaultBindingCollection;
4

```

#### *Description*

Gets the **System.Web.Services.Description.FaultBindingCollection** associated with the **System.Web.Services.Description.OperationBinding** instance.

Input

ToString

```

12 [C#] public InputBinding Input {get; set;}
13 [C++] public: __property InputBinding* get_Input();public: __property void
14 set_Input(InputBinding*);
15 [VB] Public Property Input As InputBinding
16 [JScript] public function get Input() : InputBinding;public function set
17 Input(InputBinding);
18

```

#### *Description*

Gets or sets the **System.Web.Services.Description.InputBinding** associated with the **System.Web.Services.Description.OperationBinding** instance.

An **OperationBinding** will be associated with exactly one **System.Web.Services.Description.InputBinding**.

Name

## ToString

[C#] public string Name {get; set;}

[C++] public: \_\_property String\* get\_Name();public: \_\_property void  
set\_Name(String\*);

[VB] Public Property Name As String

[JScript] public function get Name() : String;public function set Name(String);

### *Description*

Gets or sets the name of the **OperationBinding** instance.

In WSDL format, this property is realized as the **name** attribute of the tag enclosed within the tags. Note that an empty string ("" ) is returned if this property value has not been set.

### *Output*

## ToString

[C#] public OutputBinding Output {get; set;}

[C++] public: \_\_property OutputBinding\* get\_Output();public: \_\_property void  
set\_Output(OutputBinding\*);

[VB] Public Property Output As OutputBinding

[JScript] public function get Output() : OutputBinding;public function set  
Output(OutputBinding);

### *Description*

1 Gets or sets the **System.Web.Services.Description.OutputBinding**  
2 associated with the **System.Web.Services.Description.OperationBinding**  
3 instance.

4 An **OperationBinding** will be associated with exactly one  
5 **System.Web.Services.Description.OutputBinding** .

6 **OperationBindingCollection** class (**System.Web.Services.Description**)

7 **ToString**

8  
9  
10 *Description*

11 Represents a collection of  
12 **System.Web.Services.Description.OperationBinding** objects. This class cannot  
13 be inherited.

14 **Count**

15 **InnerList**

16 **Item**

17 **ToString**

18  
19  
20 *Description*

21 Gets or sets the value of a  
22 **System.Web.Services.Description.OperationBinding** at the specified zero-based  
23 index. The zero-based index of the  
24 **System.Web.Services.Description.OperationBinding** whose value is modified  
25 or returned.

List

Table

Add

[C#] public int Add(OperationBinding bindingOperation);

[C++] public: int Add(OperationBinding\* bindingOperation);

[VB] Public Function Add(ByVal bindingOperation As OperationBinding) As

Integer

[JScript] public function Add(bindingOperation : OperationBinding) : int;

#### *Description*

Adds the specified **System.Web.Services.Description.OperationBinding** instance to the end of the **OperationBindingCollection**.

*Return Value:* Returns the zero-based index where *bindingOperation* has been added. The **System.Web.Services.Description.OperationBinding** to be added to the collection.

Contains

[C#] public bool Contains(OperationBinding bindingOperation);

[C++] public: bool Contains(OperationBinding\* bindingOperation);

[VB] Public Function Contains(ByVal bindingOperation As OperationBinding)

As Boolean

[JScript] public function Contains(bindingOperation : OperationBinding) :

Boolean;

1  
2 *Description*

3 Gets a value indicating whether the specified  
4 **System.Web.Services.Description.OperationBinding** is a member of the  
5 **OperationBindingCollection** .

6 *Return Value:* **true** if *bindingOperation* is a member of the  
7 **OperationBindingCollection** ; otherwise, **false** . An  
8 **System.Web.Services.Description.OperationBinding** object.

9 CopyTo

10  
11 [C#] public void CopyTo(OperationBinding[] array, int index);  
12 [C++] public: void CopyTo(OperationBinding\* array[], int index);  
13 [VB] Public Sub CopyTo(ByVal array() As OperationBinding, ByVal index As  
14 Integer)  
15 [JScript] public function CopyTo(array : OperationBinding[], index : int);  
16

17 *Description*

18 Copies the entire **OperationBindingCollection** to a compatible one-  
19 dimensional array of type **System.Web.Services.Description.OperationBinding**  
20 , starting at the specified zero-based index of the target array. An array of type  
21 **System.Web.Services.Description.OperationBinding** that serves as the  
22 destination for the copy action. The zero-based index at which to start placing the  
23 copied collection into the target array.

24 IndexOf



```

1 [C#] public int IndexOf(OperationBinding bindingOperation);
2
3 [C++] public: int IndexOf(OperationBinding* bindingOperation);
4
5 [VB] Public Function IndexOf(ByVal bindingOperation As OperationBinding) As
6 Integer
7
8 [JScript] public function IndexOf(bindingOperation : OperationBinding) : int;

```

### *Description*

Searches for the specified **System.Web.Services.Description.OperationBinding** and returns the zero-based index of the first occurrence within the **OperationBindingCollection**.

*Return Value:* Returns a 32-bit signed integer. An

**System.Web.Services.Description.OperationBinding** object.

### *Insert*

```

14 Insert
15
16 [C#] public void Insert(int index, OperationBinding bindingOperation);
17
18 [C++] public: void Insert(int index, OperationBinding* bindingOperation);
19
20 [VB] Public Sub Insert(ByVal index As Integer, ByVal bindingOperation As
21 OperationBinding)
22
23 [JScript] public function Insert(index : int, bindingOperation : OperationBinding);

```

### *Description*

Adds the specified **System.Web.Services.Description.OperationBinding** instance to the **OperationBindingCollection** at the specified index.

1 If the number of items in the collection already equals the collection's  
2 capacity, the capacity is doubled by automatically reallocating the internal array  
3 before the new element is inserted. The zero-based index at which to insert  
4 *bindingOperation*. The **System.Web.Services.Description.OperationBinding** to  
5 be added to the collection.

#### 6 Remove

7  
8 [C#] public void Remove(OperationBinding bindingOperation);  
9 [C++] public: void Remove(OperationBinding\* bindingOperation);  
10 [VB] Public Sub Remove(ByVal bindingOperation As OperationBinding)  
11 [JScript] public function Remove(bindingOperation : OperationBinding);  
12

#### 13 Description

14 Removes the first occurrence of the specified  
15 **System.Web.Services.Description.OperationBinding** from the  
16 **OperationBindingCollection** .

17 This method performs a linear search; therefore, the average execution time  
18 is proportional to

19 **System.Web.Services.Description.OperationBindingCollection.Count** . An  
20 **System.Web.Services.Description.OperationBinding** object.

#### 21 SetParent

22  
23 [C#] protected override void SetParent(object value, object parent);  
24 [C++] protected: void SetParent(Object\* value, Object\* parent);  
25 [VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As

Object)

[JScript] protected override function SetParent(value : Object, parent : Object);

#### *Description*

Sets the parent **System.Web.Services.Description.Binding** of a member of the **OperationBindingCollection** . An object, of type **System.Web.Services.Description.OperationBinding**, within the collection. The object, of type **System.Web.Services.Description.Binding**, to be set as the parent.

OperationCollection class (System.Web.Services.Description)

ToString

#### *Description*

Represents a collection of **System.Web.Services.Description.Operation** objects. This class cannot be inherited.

Count

InnerList

Item

ToString

#### *Description*

Gets or sets the value of a **System.Web.Services.Description.Operation** at the specified zero-based index. The zero-based index of the

**System.Web.Services.Description.Operation** whose value is modified or returned.

List

Table

Add

[C#] public int Add(Operation operation);

[C++] public: int Add(Operation\* operation);

[VB] Public Function Add(ByVal operation As Operation) As Integer

[JScript] public function Add(operation : Operation) : int;

#### *Description*

Adds the specified **System.Web.Services.Description.Operation** to the end of the **OperationCollection**.

**Return Value:** Returns the zero-based index where *operation* has been added. The **System.Web.Services.Description.Operation** to be added to the **OperationCollection**.

Contains

[C#] public bool Contains(Operation operation);

[C++] public: bool Contains(Operation\* operation);

[VB] Public Function Contains(ByVal operation As Operation) As Boolean

[JScript] public function Contains(operation : Operation) : Boolean;

#### *Description*

1 Gets a value indicating whether the specified  
2 **System.Web.Services.Description.Operation** is a member of the  
3 **OperationCollection** .  
4 *Return Value:* **true** if *operation* is a member of the **OperationCollection** ;  
5 otherwise, **false** . An **System.Web.Services.Description.Operation** object.

#### 6 CopyTo

7  
8 [C#] public void CopyTo(Operation[] array, int index);  
9 [C++] public: void CopyTo(Operation\* array[], int index);  
10 [VB] Public Sub CopyTo(ByVal array() As Operation, ByVal index As Integer)  
11 [JScript] public function CopyTo(array : Operation[], index : int);  
12

#### 13 Description

14 Copies the entire **OperationCollection** to a compatible one-dimensional  
15 array of type **System.Web.Services.Description.Operation** , starting at the  
16 specified index of the target array. An array of type  
17 **System.Web.Services.Description.Operation** serving as the destination for the  
18 copy action. The zero-based index at which to start placing the copied collection.

#### 19 IndexOf

20  
21 [C#] public int IndexOf(Operation operation);  
22 [C++] public: int IndexOf(Operation\* operation);  
23 [VB] Public Function IndexOf(ByVal operation As Operation) As Integer  
24 [JScript] public function IndexOf(operation : Operation) : int;  
25

## *Description*

Searches for the specified **System.Web.Services.Description.Operation** and returns the zero-based index of the first occurrence within the **OperationCollection**.

*Return Value:* Returns a 32-bit signed integer. An **System.Web.Services.Description.Operation** object.

### Insert

[C#] public void Insert(int index, Operation operation);

[C++] public: void Insert(int index, Operation\* operation);

[VB] Public Sub Insert(ByVal index As Integer, ByVal operation As Operation)

[JScript] public function Insert(index : int, operation : Operation);

## *Description*

Adds the specified **System.Web.Services.Description.Operation** instance to the **OperationCollection** at the specified index.

If the number of items in the collection already equals the collection's capacity, the capacity is doubled by automatically reallocating the internal array before the new element is inserted. The zero-based index at which to insert *operation*. The **System.Web.Services.Description.Operation** to be added to the **OperationCollection**.

### Remove

[C#] public void Remove(Operation operation);

1 [C++] public: void Remove(Operation\* operation);

2 [VB] Public Sub Remove(ByVal operation As Operation)

3 [JScript] public function Remove(operation : Operation);

4  
5 *Description*

6 Removes the first occurrence of the specified

7 **System.Web.Services.Description.Operation** from the **OperationCollection** .

8 This method performs a linear search; therefore, the average execution time  
9 is proportional to

10 **System.Web.Services.Description.OperationCollection.Count** . The

11 **System.Web.Services.Description.Operation** object to be removed from the  
12 Collection.

13 SetParent

14  
15 [C#] protected override void SetParent(object value, object parent);

16 [C++] protected: void SetParent(Object\* value, Object\* parent);

17 [VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As  
18 Object)

19 [JScript] protected override function SetParent(value : Object, parent : Object);

20  
21 *Description*

22 Sets the parent **System.Web.Services.Description.PortType** of a member  
23 of **OperationCollection** . An object, of type  
24 **System.Web.Services.Description.Operation**, within the **OperationCollection**.

1 The object, of type **System.Web.Services.Description.PortType**, to set as the  
2 parent.

3       OperationFault class (System.Web.Services.Description)

4       ToString

5  
6  
7 *Description*

8       Defines the specifications for error messages returned by the Web Service.

9       This class cannot be inherited.

10       Instances of this class will be members of the

11 **System.Web.Services.Description.Operation.Faults** property of the parent

12 **Operation .**

13       OperationFault

14       *Example Syntax:*

15       ToString

16  
17 [C#] public OperationFault();

18 [C++] public: OperationFault();

19 [VB] Public Sub New()

20 [JScript] public function OperationFault();

21       Documentation

22       Message

23       Name

24       Operation

25       OperationFaultCollection class (System.Web.Services.Description)



1 ToString

2

3

4 *Description*

5 Represents a collection of

6 **System.Web.Services.Description.OperationFault** objects. This class cannot be

7 inherited.

8 Count

9 InnerList

10 Item

11 ToString

12 **System.Web.Services.Description.OperationFault**

13

14 *Description*

15 Gets or sets the value of a

16 **System.Web.Services.Description.OperationFault** at the specified zero-based

17 index. The zero-based index of the

18 **System.Web.Services.Description.OperationFault** whose value is modified or

19 returned.

20 Item

21 ToString

22

23 [C#] public OperationFault this[string name] {get;}

24 [C++] public: \_\_property OperationFault\* get\_Item(String\* name);

25 [VB] Public Default ReadOnly Property Item(ByVal name As String) As

1    **OperationFault**

2    [JScript] return Value = OperationFaultCollectionObject.Item(name);

4    *Description*

5       Gets an **System.Web.Services.Description.OperationFault** by its  
6    **System.Web.Services.Description.OperationFault.Name** property. The string  
7    value representing the name of the  
8    **System.Web.Services.Description.OperationFault** returned.

9       List

10      Table

11      Add

13    [C#] public int Add(OperationFault operationFaultMessage);

14    [C++] public: int Add(OperationFault\* operationFaultMessage);

15    [VB] Public Function Add(ByVal operationFaultMessage As OperationFault) As

16    Integer

17    [JScript] public function Add(operationFaultMessage : OperationFault) : int;

19    *Description*

20       Adds the specified **System.Web.Services.Description.OperationFault** to  
21    the end of the **OperationFaultCollection** .

22    *Return Value:* Returns the zero-based index where *operationFaultMessage* has  
23    been added. The **System.Web.Services.Description.OperationFault** to be added  
24    to the **OperationFaultCollection**.

25       Contains

```

1 [C#] public bool Contains(OperationFault operationFaultMessage);
2
3 [C++] public: bool Contains(OperationFault* operationFaultMessage);
4
5 [VB] Public Function Contains(ByVal operationFaultMessage As OperationFault)
6 As Boolean
7
8 [JScript] public function Contains(operationFaultMessage : OperationFault) :
9 Boolean;
10

```

#### *Description*

Gets a value indicating whether the specified **System.Web.Services.Description.OperationFault** is a member of the **OperationFaultCollection**.

*Return Value:* **true** if *operationFaultMessage* is a member of the **OperationFaultCollection**; otherwise, **false**. An **System.Web.Services.Description.OperationFault** object.

#### *CopyTo*

```

16
17
18 [C#] public void CopyTo(OperationFault[] array, int index);
19
20 [C++] public: void CopyTo(OperationFault* array[], int index);
21
22 [VB] Public Sub CopyTo(ByVal array() As OperationFault, ByVal index As
23 Integer)
24
25 [JScript] public function CopyTo(array : OperationFault[], index : int);

```

#### *Description*

Copies the entire **System.Web.Services.Description.OperationFaultCollection** to a compatible one-dimensional array, starting at the specified zero-based index of the target array. An array of type **System.Web.Services.Description.OperationFault** serving as the destination of the copy action. The zero-based index at which to start placing the copied collection.

#### GetKey

[C#] protected override string GetKey(object value);

[C++] protected: String\* GetKey(Object\* value);

[VB] Overrides Protected Function GetKey(ByVal value As Object) As String

[JScript] protected override function GetKey(value : Object) : String;

#### *Description*

Returns the name of the **System.Web.Services.Description.OperationFault** associated with the value passed by reference. An object for which to return the name.

#### IndexOf

[C#] public int IndexOf(OperationFault operationFaultMessage);

[C++] public: int IndexOf(OperationFault\* operationFaultMessage);

[VB] Public Function IndexOf(ByVal operationFaultMessage As OperationFault)  
As Integer

[JScript] public function IndexOf(operationFaultMessage : OperationFault) : int;

## *Description*

Searches for the specified **System.Web.Services.Description.OperationFault** and returns the zero-based index of the first occurrence within the collection.

*Return Value:* Returns a 32-bit signed integer. An **System.Web.Services.Description.OperationFault** object.

## *Insert*

```
[C#] public void Insert(int index, OperationFault operationFaultMessage);  
[C++] public: void Insert(int index, OperationFault* operationFaultMessage);  
[VB] Public Sub Insert(ByVal index As Integer, ByVal operationFaultMessage As  
OperationFault)  
[JScript] public function Insert(index : int, operationFaultMessage :  
OperationFault);
```

## *Description*

Adds the specified **System.Web.Services.Description.OperationFault** to the **OperationFaultCollection** at the specified zero-based index.

If the number of items in the collection already equals the collection's capacity, the capacity is doubled by automatically reallocating the internal array before the new element is inserted. The zero-based index at which to insert *operationFaultMessage*. The **System.Web.Services.Description.OperationFault** to be added to the Collection.

## *Remove*

```

1
2 [C#] public void Remove(OperationFault operationFaultMessage);
3 [C++] public: void Remove(OperationFault* operationFaultMessage);
4 [VB] Public Sub Remove(ByVal operationFaultMessage As OperationFault)
5 [JScript] public function Remove(operationFaultMessage : OperationFault);
6

```

### *Description*

Removes the first occurrence of the specified **System.Web.Services.Description.OperationFault** from the **OperationFaultCollection**.

This method performs a linear search; therefore, the average execution time is proportional to

**System.Web.Services.Description.OperationFaultCollection.Count**. The **System.Web.Services.Description.OperationFault** object to be removed from the collection.

### **SetParent**

```

17
18 [C#] protected override void SetParent(object value, object parent);
19 [C++] protected: void SetParent(Object* value, Object* parent);
20 [VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As
21 Object)
22 [JScript] protected override function SetParent(value : Object, parent : Object);
23

```

### *Description*

1       Sets the parent **System.Web.Services.Description.Operation** of a member  
2 of the **OperationFaultCollection** . An object, of class  
3 **System.Web.Services.Description.OperationFault** , within the collection. The  
4 object, of class **System.Web.Services.Description.Operation** , to be set as the  
5 parent.

6       OperationFlow enumeration (System.Web.Services.Description)

7       ToString

8  
9  
10    *Description*

11       Specifies the type of transmission an endpoint of the Web Service can  
12 support.

13       Although Request-Response or Solicit-Response are logically correlated in  
14 the WSDL document, the concrete correlation information will be specified by a  
15 binding. For example, the request and response messages may be exchanged as  
16 part of one or two actual HTTP transmissions.

17       ToString

18  
19    [C#] public const OperationFlow None;

20    [C++] public: const OperationFlow None;

21    [VB] Public Const None As OperationFlow

22    [JScript] public var None : OperationFlow;

23  
24    *Description*

25       Indicates that the endpoint of the Web Service receives no transmissions.

ToString

[C#] public const OperationFlow Notification;  
[C++] public: const OperationFlow Notification;  
[VB] Public Const Notification As OperationFlow  
[JScript] public var Notification : OperationFlow;

*Description*

Indicates that an endpoint of the Web Service sends a message.

ToString

[C#] public const OperationFlow OneWay;  
[C++] public: const OperationFlow OneWay;  
[VB] Public Const OneWay As OperationFlow  
[JScript] public var OneWay : OperationFlow;

*Description*

Indicates that the endpoint of the Web Service receives a message.

ToString

[C#] public const OperationFlow RequestResponse;  
[C++] public: const OperationFlow RequestResponse;  
[VB] Public Const RequestResponse As OperationFlow  
[JScript] public var RequestResponse : OperationFlow;



1  
2 *Description*

3 Indicates that an endpoint of the Web Service receives a message, then  
4 sends a correlated message.

5 ToString

6  
7 [C#] public const OperationFlow SolicitResponse;

8 [C++] public: const OperationFlow SolicitResponse;

9 [VB] Public Const SolicitResponse As OperationFlow

10 [JScript] public var SolicitResponse : OperationFlow;

11  
12 *Description*

13 Indicates that an endpoint of the Web Service sends a message, then  
14 receives a correlated message.

15 OperationInput class (System.Web.Services.Description)

16 ToString

17  
18  
19 *Description*

20 Defines the specifications for input messages received by the Web Service.  
21 This class cannot be inherited.

22 Exactly one instance of this class will be a member of the  
23 **System.Web.Services.Description.Operation.Messages** property of the parent  
24 **System.Web.Services.Description.Operation** .

25 OperationInput

*Example Syntax:*

ToString

[C#] public OperationInput();

[C++] public: OperationInput();

[VB] Public Sub New()

[JScript] public function OperationInput();

Documentation

Message

Name

Operation

OperationMessage class (System.Web.Services.Description)

ToString

## *Description*

Serves as the base class for the following classes:

**System.Web.Services.Description.OperationFaultSystem.Web.Services.Descri  
ption.OperationInputSystem.Web.Services.Description.OperationOutput**

In its default implementation, an **OperationMessage** instance refers to any message type passed by the action of a service.

OperationMessage

*Example Syntax:*

ToString

1  
2 [C#] protected OperationMessage();  
3 [C++] protected: OperationMessage();  
4 [VB] Protected Sub New()  
5 [JScript] protected function OperationMessage();

6       Documentation

7       Message

8       ToString

9

10

11       *Description*

12       Gets or sets an abstract, typed definition of the data being communicated.

13       Name

14       ToString

15

16 [C#] public string Name {get; set;}

17 [C++] public: \_\_property String\* get\_Name();public: \_\_property void

18 set\_Name(String\*);

19 [VB] Public Property Name As String

20 [JScript] public function get Name() : String;public function set Name(String);

21

22       *Description*

23       The name of the **System.Web.Services.Description.OperationMessage**  
24 instance.

25       Operation

ToString

[C#] public Operation Operation {get;}

[C++] public: \_\_property Operation\* get\_Operation();

[VB] Public ReadOnly Property Operation As Operation

[JScript] public function get Operation() : Operation;

### *Description*

Gets the **System.Web.Services.Description.Operation** of which the **System.Web.Services.Description.OperationMessage** instance is a member.

OperationMessageCollection class (System.Web.Services.Description)

ToString

### *Description*

Represents a collection of **System.Web.Services.Description.OperationInput** and **System.Web.Services.Description.OperationOutput** messages related to a Web Service. This class cannot be inherited.

An instance of this class will be returned by the **System.Web.Services.Description.Operation.Messages** property of the parent **System.Web.Services.Description.Operation** . As such, it can have exactly two members, one an **System.Web.Services.Description.OperationInput** and the other an **System.Web.Services.Description.OperationOutput** .

Count

1 Flow  
2 ToString

3  
4  
5 *Description*

6 Gets the type of transmission supported by the  
7 **OperationMessageCollection** .

8 InnerList  
9 Input  
10 ToString

11  
12  
13 *Description*

14 Gets the first occurrence of an  
15 **System.Web.Services.Description.OperationInput** instance within the  
16 collection.

17 Th  
18 Item  
19 ToString

20  
21 [C#] public OperationMessage this[int index] {get; set;}  
22 [C++] public: \_\_property OperationMessage\* get\_Item(int index);public:  
23 \_\_property void set\_Item(int index, OperationMessage\*);  
24 [VB] Public Default Property Item(ByVal index As Integer) As OperationMessage  
25 [JScript] returnValue =

1 OperationMessageCollectionObject.Item(index);OperationMessageCollectionObje  
2 ct.Item(index) = returnValue;

3  
4 *Description*

5 Gets or sets the value of an  
6 **System.Web.Services.Description.OperationMessage** at the specified zero-  
7 based index. The zero-based index of the  
8 **System.Web.Services.Description.OperationMessage** whose value is modified  
9 or returned.

10 List

11 Output

12 ToString

13  
14  
15 *Description*

16 Gets the first occurrence of an  
17 **System.Web.Services.Description.OperationOutput** within the collection.  
18 The search is performed in index order; thus the property returns only the  
19 **System.Web.Services.Description.OperationOutput** with the lowest-numbered  
20 index.

21 Table

22 Add

23  
24 [C#] public int Add(OperationMessage operationMessage);

25 [C++] public: int Add(OperationMessage\* operationMessage);

1 [VB] Public Function Add(ByVal operationMessage As OperationMessage) As  
2 Integer

3 [JScript] public function Add(operationMessage : OperationMessage) : int;

4  
5 *Description*

6 Adds the specified **System.Web.Services.Description.OperationMessage**  
7 to the end of the **OperationMessageCollection** .

8 *Return Value:* Returns the zero-based index where *operationMessage* has been  
9 added. The **System.Web.Services.Description.OperationMessage** to be added to  
10 the collection.

11 Contains

12  
13 [C#] public bool Contains(OperationMessage operationMessage);

14 [C++] public: bool Contains(OperationMessage\* operationMessage);

15 [VB] Public Function Contains(ByVal operationMessage As OperationMessage)

16 As Boolean

17 [JScript] public function Contains(operationMessage : OperationMessage) :

18 Boolean;

19  
20 *Description*

21 Gets a value indicating whether the specified  
22 **System.Web.Services.Description.OperationMessage** instance is a member of  
23 the **OperationMessageCollection** .

24 *Return Value:* **true** if *operationMessage* is a member of the  
25

1 **OperationMessageCollection** ; otherwise, **false** . An

2 **System.Web.Services.Description.OperationMessage** object.

3 **CopyTo**

4  
5 [C#] public void CopyTo(OperationMessage[] array, int index);

6 [C++] public: void CopyTo(OperationMessage\* array[], int index);

7 [VB] Public Sub CopyTo(ByVal array() As OperationMessage, ByVal index As  
8 Integer)

9 [JScript] public function CopyTo(array : OperationMessage[], index : int);

10  
11 *Description*

12 Copies the entire **OperationMessageCollection** to a compatible one-  
13 dimensional array of type **System.Web.Services.Description.OperationMessage**  
14 , starting at the specified zero-based index of the target array. An array of type  
15 **System.Web.Services.Description.OperationMessage** serving as the destination  
16 of the copy action. The zero-based index at which to start placing the copied  
17 collection.

18 **IndexOf**

19  
20 [C#] public int IndexOf(OperationMessage operationMessage);

21 [C++] public: int IndexOf(OperationMessage\* operationMessage);

22 [VB] Public Function IndexOf(ByVal operationMessage As OperationMessage)  
23 As Integer

24 [JScript] public function IndexOf(operationMessage : OperationMessage) : int;

25



## Description

Searches for the specified `OperationMessage` instance and returns the zero-based index of the first occurrence within the **`OperationMessageCollection`**.

*Return Value:* Returns a 32-bit signed integer. An

**`System.Web.Services.Description.OperationMessage`** object.

## Insert

[C#] `public void Insert(int index, OperationMessage operationMessage);`

[C++] `public: void Insert(int index, OperationMessage* operationMessage);`

[VB] `Public Sub Insert(ByVal index As Integer, ByVal operationMessage As OperationMessage)`

[JScript] `public function Insert(index : int, operationMessage : OperationMessage);`

## Description

Adds the specified **`System.Web.Services.Description.OperationMessage`** to the **`OperationMessageCollection`** at the specified zero-based index.

If the number of items in the collection already equals the collection's capacity, the capacity is doubled by automatically reallocating the internal array before the new element is inserted. The zero-based index at which to insert *operationMessage*. The **`System.Web.Services.Description.OperationMessage`** to be added to the **`OperationMessageCollection`**.

## OnInsert

1  
2 [C#] protected override void OnInsert(int index, object value);

3 [C++] protected: void OnInsert(int index, Object\* value);

4 [VB] Overrides Protected Sub OnInsert(ByVal index As Integer, ByVal value As  
5 Object)

6 [JScript] protected override function OnInsert(index : int, value : Object);

7  
8 *Description*

9 Performs validation upon insertion of an  
10 **System.Web.Services.Description.OperationMessage** at the specified specified.

11 This method overrides the base method in order to limit the number of  
12 collection members to two (an  
13 **System.Web.Services.Description.OperationInput** and an  
14 **System.Web.Services.Description.OperationOutput** ). The zero-based index at  
15 which to insert the specified  
16 **System.Web.Services.Description.OperationMessage**. The  
17 **System.Web.Services.Description.OperationMessage** to be added to the  
18 **OperationMessageCollection**.

19 **OnSet**

20  
21 [C#] protected override void OnSet(int index, object oldValue, object newValue);

22 [C++] protected: void OnSet(int index, Object\* oldValue, Object\* newValue);

23 [VB] Overrides Protected Sub OnSet(ByVal index As Integer, ByVal oldValue As  
24 Object, ByVal newValue As Object)

25 [JScript] protected override function OnSet(index : int, oldValue : Object,

newValue : Object);

### Description

Performs validation upon replacement of the specified member of the `OperationMessageCollection` with a new `OperationMessage` at the specified zero-based index.

This method overrides the base method in order to assure that the value replacing the existing value is of the same class as that existing value. Thus an **System.Web.Services.Description.OperationInput** must be replaced by an **OperationInput** , and an **System.Web.Services.Description.OperationOutput** must be replaced by an **OperationOutput** . The zero-based index at which to change *oldValue* to *newValue*. The object value to be replaced by *newValue*. The object value with which to replace *oldValue*.

### OnValidate

[C#] protected override void OnValidate(object value);

[C++] protected: void OnValidate(Object\* value);

[VB] Overrides Protected Sub OnValidate(ByVal value As Object)

[JScript] protected override function OnValidate(value : Object);

### Description

Performs a check on the type of the specified object when validating it.

This method overrides the base method in order to assure that the instance represented by the *value* parameter is either of type **System.Web.Services.Description.OperationInput** or

**System.Web.Services.Description.OperationOutput** . The object to be validated.

Remove

[C#] public void Remove(OperationMessage operationMessage);

[C++] public: void Remove(OperationMessage\* operationMessage);

[VB] Public Sub Remove(ByVal operationMessage As OperationMessage)

[JScript] public function Remove(operationMessage : OperationMessage);

#### *Description*

Removes the first occurrence of the specified

**System.Web.Services.Description.OperationMessage** from the **OperationMessageCollection** .

This method performs a linear search; therefore, the average execution time is proportional to **Count** . An

**System.Web.Services.Description.OperationMessage** object.

SetParent

[C#] protected override void SetParent(object value, object parent);

[C++] protected: void SetParent(Object\* value, Object\* parent);

[VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As Object)

[JScript] protected override function SetParent(value : Object, parent : Object);

#### *Description*

1       Sets the parent **System.Web.Services.Description.Operation** of the  
2 **OperationMessageCollection** . An object, of type  
3 **System.Web.Services.Description.OperationMessage** or a derived type, within  
4 the collection. The object, of type **System.Web.Services.Description.Operation** ,  
5 to be set as the parent.

6       OperationOutput class (System.Web.Services.Description)

7       ToString

8  
9  
10 *Description*

11       Defines the specifications for output messages returned by the Web  
12 Service. This class cannot be inherited.

13       Exactly one instance of this class will be a member of the  
14 **System.Web.Services.Description.Operation.Messages** property of the parent  
15 **System.Web.Services.Description.Operation** .

16       OperationOutput

17       *Example Syntax:*

18       ToString

19  
20 [C#] public OperationOutput();

21 [C++] public: OperationOutput();

22 [VB] Public Sub New()

23 [JScript] public function OperationOutput();

24       Documentation

25       Message

1 Name  
2 Operation  
3 OutputBinding class (System.Web.Services.Description)  
4 ToString  
5  
6

7 *Description*

8 Provides a set of specifications for data formats and protocols used by the  
9 Web Service for output messages. This class cannot be inherited.

10 OutputBinding

11 *Example Syntax:*

12 ToString  
13

14 [C#] public OutputBinding();

15 [C++] public: OutputBinding();

16 [VB] Public Sub New()

17 [JScript] public function OutputBinding();

18 Documentation

19 Extensions

20 ToString  
21

22  
23 *Description*  
24  
25

1 Gets the  
2 **System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect**  
3 **ion** associated with the **OutputBinding** instance.

4 Name

5 OperationBinding

6 Port class (System.Web.Services.Description)

7 ToString

8  
9  
10 *Description*

11 Defines an individual endpoint contained in the Web Service. This class  
12 cannot be inherited.

13 This class specifies a single address (URI) for a binding, which defines  
14 message format and protocol details for operations and messages for a given

15 **System.Web.Services.Description.PortType** . The

16 **System.Web.Services.Description.Port.Binding** property gets or sets those  
17 values for the **Port** instance.

18 Port

19 *Example Syntax:*

20 ToString

21  
22 [C#] public Port();

23 [C++] public: Port(); .

24 [VB] Public Sub New()

25 [JScript] public function Port();

Binding

ToString

[C#] public XmlQualifiedName Binding {get; set;}

[C++] public: \_\_property XmlQualifiedName\* get\_Binding();public: \_\_property  
void set\_Binding(XmlQualifiedName\*);

[VB] Public Property Binding As XmlQualifiedName

[JScript] public function get Binding() : XmlQualifiedName;public function set  
Binding(XmlQualifiedName);

### *Description*

Gets or sets the value of the XML binding attribute of the **Port** .

A binding defines message format and protocol details for operations and  
messages for a given **System.Web.Services.Description.PortType** . This  
property gets or sets those values for a specific **Port** .

Documentation

Extensions

ToString

### *Description*

Gets the collection of extensibility elements contained in the **Port** .

Name

ToString



1  
2 [C#] public string Name {get; set;}

3 [C++] public: \_\_property String\* get\_Name();public: \_\_property void

4 set\_Name(String\*);

5 [VB] Public Property Name As String

6 [JScript] public function get Name() : String;public function set Name(String);

7  
8 *Description*

9 Gets or sets the Name of the **Port** .

10 Service

11 ToString

12  
13 [C#] public Service Service {get;}

14 [C++] public: \_\_property Service\* get\_Service();

15 [VB] Public ReadOnly Property Service As Service

16 [JScript] public function get Service() : Service;

17  
18 *Description*

19 Gets the **System.Web.Services.Description.Service** of which the **Port**  
20 instance is a member.

21 PortCollection class (System.Web.Services.Description)

22 ToString

23  
24  
25 *Description*

Represents the collection of all **System.Web.Services.Description.Port** objects contained within a **System.Web.Services.Description.Service** . This class cannot be inherited.

Count

InnerList

Item

ToString

**System.Web.Services.Description.Port**

#### *Description*

Gets or sets the value of a **System.Web.Services.Description.Port** at the specified zero-based index. The zero-based index of the **System.Web.Services.Description.Port** whose value is modified or returned.

Item

ToString

[C#] public Port this[string name] {get;}

[C++] public: \_\_property Port\* get\_Item(String\* name);

[VB] Public Default ReadOnly Property Item(ByVal name As String) As Port

[JScript] returnValue = PortCollectionObject.Item(name);

#### *Description*

Gets the specified **System.Web.Services.Description.Port** by its **System.Web.Services.Description.Port.Name** property. A string value representing the name of the **System.Web.Services.Description.Port** returned.

List

Table

Add

[C#] public int Add(Port port);

[C++] public: int Add(Port\* port);

[VB] Public Function Add(ByVal port As Port) As Integer

[JScript] public function Add(port : Port) : int;

*Description*

Adds the specified **System.Web.Services.Description.Port** instance to the end of the **PortCollection**.

*Return Value:* Returns the index where the specified

**System.Web.Services.Description.Port** instance has been added. The

**System.Web.Services.Description.Port** to be added to the **PortCollection**.

Contains

[C#] public bool Contains(Port port);

[C++] public: bool Contains(Port\* port);

[VB] Public Function Contains(ByVal port As Port) As Boolean

[JScript] public function Contains(port : Port) : Boolean;

*Description*

Gets a value indicating whether the specified

**System.Web.Services.Description.Port** instance is a member of the

## PortCollection .

*Return Value:* **true** if the specified **System.Web.Services.Description.Port** is a member of the **PortCollection** ; otherwise, **false** . A

**System.Web.Services.Description.Port** object.

### CopyTo

[C#] public void CopyTo(Port[] array, int index);

[C++] public: void CopyTo(Port\* array[], int index);

[VB] Public Sub CopyTo(ByVal array() As Port, ByVal index As Integer)

[JScript] public function CopyTo(array : Port[], index : int);

### Description

Copies the entire **System.Web.Services.Description.PortCollection** to a one-dimensional array of type **System.Web.Services.Description.Port** , starting at the specified zero-based index of the target array. An array of type **System.Web.Services.Description.Port** serving as the destination of the copy action. The zero-based index at which to start placing the copied **PortCollection** .

### GetKey

[C#] protected override string GetKey(object value);

[C++] protected: String\* GetKey(Object\* value);

[VB] Overrides Protected Function GetKey(ByVal value As Object) As String

[JScript] protected override function GetKey(value : Object) : String;

### Description

1 Returns the name of the **System.Web.Services.Description.Port** instance  
2 associated with the value passed by reference. A

3 **System.Web.Services.Description.Port** instance for which to return the name.

4 **IndexOf**

5  
6 [C#] public int IndexOf(Port port);

7 [C++] public: int IndexOf(Port\* port);

8 [VB] Public Function IndexOf(ByVal port As Port) As Integer

9 [JScript] public function IndexOf(port : Port) : int;

10  
11 *Description*

12 Searches for the specified **System.Web.Services.Description.Port**  
13 instance and returns the zero-based index of the first occurrence within the  
14 **PortCollection** .

15 *Return Value:* Returns a 32-bit signed integer. A

16 **System.Web.Services.Description.Port** object.

17 **Insert**

18  
19 [C#] public void Insert(int index, Port port);

20 [C++] public: void Insert(int index, Port\* port);

21 [VB] Public Sub Insert(ByVal index As Integer, ByVal port As Port)

22 [JScript] public function Insert(index : int, port : Port);

23  
24 *Description*

25

1 Adds the specified **System.Web.Services.Description.Port** instance to the  
2 **PortCollection** at the specified index.

3 If the number of items in the collection already equals the collection's  
4 capacity, the capacity is doubled by automatically reallocating the internal array  
5 before the new element is inserted. The zero-based index at which to insert *port*.  
6 The **System.Web.Services.Description.Port** to be added to the collection.

#### 7 Remove

8  
9 [C#] public void Remove(Port port);

10 [C++] public: void Remove(Port\* port);

11 [VB] Public Sub Remove(ByVal port As Port)

12 [JScript] public function Remove(port : Port);

#### 13 14 *Description*

15 Removes the first occurrence of the specified  
16 **System.Web.Services.Description.Port** from the **PortCollection** .

17 This method performs a linear search; therefore, the average execution time  
18 is proportional to **Count** . A **System.Web.Services.Description.Port** object.

#### 19 SetParent

20  
21 [C#] protected override void SetParent(object value, object parent);

22 [C++] protected: void SetParent(Object\* value, Object\* parent);

23 [VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As  
24 Object)

25 [JScript] protected override function SetParent(value : Object, parent : Object);

1  
2 *Description*

3       Sets the parent **System.Web.Services.Description.Service** of a member of  
4 the **PortCollection** . An object, of type **System.Web.Services.Description.Port**,  
5 within the collection. The object, of type  
6 **System.Web.Services.Description.Service** , to set as the parent.

7       PortType class (System.Web.Services.Description)

8       ToString

9  
10  
11 *Description*

12       Represents a named set of abstract operations and the corresponding  
13 abstract messages. This class cannot be inherited.

14       PortType

15       *Example Syntax:*

16       ToString

17  
18 [C#] public PortType();

19 [C++] public: PortType();

20 [VB] Public Sub New()

21 [JScript] public function PortType();

22       Documentation

23       Name

24       ToString

1  
2  
3 *Description*

4 Gets or sets the name of the **System.Web.Services.Description.PortType**  
5 instance.

6 Returns an empty string ("" ) if the property value has not been assigned.

7 Operations

8 ToString

9  
10 [C#] public OperationCollection Operations {get;}

11 [C++] public: \_\_property OperationCollection\* get\_Operations();

12 [VB] Public ReadOnly Property Operations As OperationCollection

13 [JScript] public function get Operations() : OperationCollection;

14  
15 *Description*

16 Gets the collection of **System.Web.Services.Description.Operation**  
17 instances defined by the **System.Web.Services.Description.PortType** instance.

18 ServiceDescription

19 ToString

20  
21 [C#] public ServiceDescription ServiceDescription {get;}

22 [C++] public: \_\_property ServiceDescription\* get\_ServiceDescription();

23 [VB] Public ReadOnly Property ServiceDescription As ServiceDescription

24 [JScript] public function get ServiceDescription() : ServiceDescription;



1  
2 *Description*

3 Gets the **System.Web.Services.Description.ServiceDescription** of which  
4 the **System.Web.Services.Description.PortType** instance is a member.

5 PortTypeCollection class (System.Web.Services.Description)

6 ToString

7  
8  
9 *Description*

10 Represents a collection of **System.Web.Services.Description.PortType**  
11 objects, that is, a collection of sets of operations supported by the Web Service.

12 This class cannot be inherited.

13 Count

14 InnerList

15 Item

16 ToString

17 **System.Web.Services.Description.PortType**

18  
19 *Description*

20 Gets or sets the value of a **System.Web.Services.Description.PortType** at  
21 the specified zero-based index. The zero-based index of the  
22 **System.Web.Services.Description.PortType** whose value is modified or  
23 returned.

24 Item

25 ToString

1  
2 [C#] public PortType this[string name] {get;}

3 [C++] public: \_\_property PortType\* get\_Item(String\* name);

4 [VB] Public Default ReadOnly Property Item(ByVal name As String) As

5 PortType

6 [JScript] returnValue = PortTypeCollectionObject.Item(name);

7  
8 *Description*

9 Gets the specified **System.Web.Services.Description.PortType** by its  
10 **System.Web.Services.Description.PortType.Name** property.

11 This method performs a linear search; therefore, the average execution time  
12 is proportional to **System.Web.Services.Description.PortTypeCollection.Count**  
13 . The string value representing the name of the  
14 **System.Web.Services.Description.PortType** returned.

15 List

16 Table

17 Add

18  
19 [C#] public int Add(PortType portType);

20 [C++] public: int Add(PortType\* portType);

21 [VB] Public Function Add(ByVal portType As PortType) As Integer

22 [JScript] public function Add(portType : PortType) : int;

23  
24 *Description*

1 Adds the specified **System.Web.Services.Description.PortType** to the end  
2 of the **PortTypeCollection** .

3 *Return Value:* Returns the index where the

4 **System.Web.Services.Description.PortType** instance specified by the *portType*  
5 parameter has been added. The **System.Web.Services.Description.PortType** to  
6 be added to the **PortTypeCollection**.

7 Contains

9 [C#] public bool Contains(PortType portType);

10 [C++] public: bool Contains(PortType\* portType);

11 [VB] Public Function Contains(ByVal portType As PortType) As Boolean

12 [JScript] public function Contains(portType : PortType) : Boolean;

14 *Description*

15 Gets a value indicating whether the specified  
16 **System.Web.Services.Description.PortType** instance is a member of the  
17 **PortTypeCollection** .

18 *Return Value:* **true** if the specified **System.Web.Services.Description.PortType**  
19 instance is a member of the **PortTypeCollection** ; otherwise, **false** . A  
20 **System.Web.Services.Description.PortType** object.

21 CopyTo

23 [C#] public void CopyTo(PortType[] array, int index);

24 [C++] public: void CopyTo(PortType\* array[], int index);

25 [VB] Public Sub CopyTo(ByVal array() As PortType, ByVal index As Integer)

1 [JScript] public function CopyTo(array : PortType[], index : int);

2  
3 *Description*

4 Copies the entire **System.Web.Services.Description.PortTypeCollection**  
5 to a one-dimensional array of type PortType, starting at the specified zero-based  
6 index of the target array. An array of type  
7 **System.Web.Services.Description.PortType** serving as the destination of the  
8 copy action. The zero-based index at which to start placing the copied  
9 **PortTypeCollection**.

10 **GetKey**

11  
12 [C#] protected override string GetKey(object value);

13 [C++] protected: String\* GetKey(Object\* value);

14 [VB] Overrides Protected Function GetKey(ByVal value As Object) As String

15 [JScript] protected override function GetKey(value : Object) : String;

16  
17 *Description*

18 Returns the name of the **System.Web.Services.Description.PortType**  
19 associated with the value passed by reference. An object for which to return the  
20 name.

21 **IndexOf**

22  
23 [C#] public int IndexOf(PortType portType);

24 [C++] public: int IndexOf(PortType\* portType);

25 [VB] Public Function IndexOf(ByVal portType As PortType) As Integer

1 [JScript] public function IndexOf(portType : PortType) : int;

3 *Description*

4 Searches for the specified **System.Web.Services.Description.PortType**  
5 instance and returns the zero-based index of the first occurrence within the  
6 **PortTypeCollection**.

7 *Return Value:* Returns a 32-bit signed integer. A  
8 **System.Web.Services.Description.PortType** object.

9 Insert

11 [C#] public void Insert(int index, PortType portType);

12 [C++] public: void Insert(int index, PortType\* portType);

13 [VB] Public Sub Insert(ByVal index As Integer, ByVal portType As PortType)

14 [JScript] public function Insert(index : int, portType : PortType);

16 *Description*

17 Adds the specified **System.Web.Services.Description.PortType** instance  
18 to the **PortTypeCollection** at the specified zero-based index.

19 If the number of items in the collection already equals the collection's  
20 capacity, the capacity is doubled by automatically reallocating the internal array  
21 before the new element is inserted. The zero-based index at which to insert the  
22 specified **System.Web.Services.Description.PortType** instance. The  
23 **System.Web.Services.Description.PortType** to be added to the  
24 **PortTypeCollection**.

25 Remove

```

1
2 [C#] public void Remove(PortType portType);
3 [C++] public: void Remove(PortType* portType);
4 [VB] Public Sub Remove(ByVal portType As PortType)
5 [JScript] public function Remove(portType : PortType);
6

```

### *Description*

Removes the first occurrence of the specified **System.Web.Services.Description.PortType** from the **PortTypeCollection** .

This method performs a linear search; therefore, the average execution time is proportional to **Count** . A **System.Web.Services.Description.PortType** object.

### **SetParent**

```

14 [C#] protected override void SetParent(object value, object parent);
15 [C++] protected: void SetParent(Object* value, Object* parent);
16 [VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As
17 Object)
18 [JScript] protected override function SetParent(value : Object, parent : Object);
19

```

### *Description*

Sets the parent **System.Web.Services.Description.ServiceDescription** of a member of **System.Web.Services.Description.PortTypeCollection** . An object, of type **System.Web.Services.Description.PortType**, within the **PortTypeCollection** . The object, of type **System.Web.Services.Description.ServiceDescription** , to set as the parent.

1       ProtocolImporter class (System.Web.Services.Description)  
2       ToString

3

4

5       *Description*

6

7       ProtocolImporter

8       *Example Syntax:*

9       ToString

10

11      [C#] protected ProtocolImporter();

12      [C++] protected: ProtocolImporter();

13      [VB] Protected Sub New()

14      [JScript] protected function ProtocolImporter();

15       AbstractSchemas

16       ToString

17

18      [C#] public XmlSchemas AbstractSchemas {get;}

19      [C++] public: \_\_property XmlSchemas\* get\_AbstractSchemas();

20      [VB] Public ReadOnly Property AbstractSchemas As XmlSchemas

21      [JScript] public function get AbstractSchemas() : XmlSchemas;

22

23       *Description*

24

25       Binding

ToString

[C#] public Binding Binding {get;}

[C++] public: \_\_property Binding\* get\_Binding();

[VB] Public ReadOnly Property Binding As Binding

[JScript] public function get Binding() : Binding;

*Description*

ClassName

ToString

[C#] public string ClassName {get;}

[C++] public: \_\_property String\* get\_ClassName();

[VB] Public ReadOnly Property ClassName As String

[JScript] public function get ClassName() : String;

*Description*

ClassNames

ToString

[C#] public CodeIdentifiers ClassNames {get;}

[C++] public: \_\_property CodeIdentifiers\* get\_ClassNames();

[VB] Public ReadOnly Property ClassNames As CodeIdentifiers



1 [JScript] public function get ClassNames() : CodeIdentifiers;

2  
3 *Description*

4  
5 CodeNamespace

6 ToString

7  
8 [C#] public CodeNamespace CodeNamespace {get;}

9 [C++] public: \_\_property CodeNamespace\* get\_CodeNamespace();

10 [VB] Public ReadOnly Property CodeNamespace As CodeNamespace

11 [JScript] public function get CodeNamespace() : CodeNamespace;

12  
13 *Description*

14  
15 CodeTypeDeclaration

16 ToString

17  
18 [C#] public CodeTypeDeclaration CodeTypeDeclaration {get;}

19 [C++] public: \_\_property CodeTypeDeclaration\* get\_CodeTypeDeclaration();

20 [VB] Public ReadOnly Property CodeTypeDeclaration As CodeTypeDeclaration

21 [JScript] public function get CodeTypeDeclaration() : CodeTypeDeclaration;

22  
23 *Description*

24  
25 ConcreteSchemas

ToString

```
[C#] public XmlSchemas ConcreteSchemas {get;}
[C++] public: __property XmlSchemas* get_ConcreteSchemas();
[VB] Public ReadOnly Property ConcreteSchemas As XmlSchemas
[JScript] public function get ConcreteSchemas() : XmlSchemas;
```

*Description*

InputMessage

ToString

```
[C#] public Message InputMessage {get;}
[C++] public: __property Message* get_InputMessage();
[VB] Public ReadOnly Property InputMessage As Message
[JScript] public function get InputMessage() : Message;
```

*Description*

MethodName

ToString

```
[C#] public string MethodName {get;}
[C++] public: __property String* get_MethodName();
[VB] Public ReadOnly Property MethodName As String
```

1 [JScript] public function get MethodName() : String;

2  
3 *Description*

4  
5       Operation

6       ToString

7  
8 [C#] public Operation Operation {get;}

9 [C++] public: \_\_property Operation\* get\_Operation();

10 [VB] Public ReadOnly Property Operation As Operation

11 [JScript] public function get Operation() : Operation;

12  
13 *Description*

14  
15       OperationBinding

16       ToString

17  
18 [C#] public OperationBinding OperationBinding {get;}

19 [C++] public: \_\_property OperationBinding\* get\_OperationBinding();

20 [VB] Public ReadOnly Property OperationBinding As OperationBinding

21 [JScript] public function get OperationBinding() : OperationBinding;

22  
23 *Description*

24  
25       OutputMessage

ToString

```
[C#] public Message OutputMessage {get;}
[C++] public: __property Message* get_OutputMessage();
[VB] Public ReadOnly Property OutputMessage As Message
[JScript] public function get OutputMessage() : Message;
```

*Description*

Port

ToString

```
[C#] public Port Port {get;}
[C++] public: __property Port* get_Port();
[VB] Public ReadOnly Property Port As Port
[JScript] public function get Port() : Port;
```

*Description*

PortType

ToString

```
[C#] public PortType PortType {get;}
[C++] public: __property PortType* get_PortType();
[VB] Public ReadOnly Property PortType As PortType
```



1 ToString

2

3 [C#] public Service Service {get;}

4 [C++] public: \_\_property Service\* get\_Service();

5 [VB] Public ReadOnly Property Service As Service

6 [JScript] public function get Service() : Service;

7

8 *Description*

9

10 ServiceDescriptions

11 ToString

12

13 [C#] public ServiceDescriptionCollection ServiceDescriptions {get;}

14 [C++] public: \_\_property ServiceDescriptionCollection\*

15 get\_ServiceDescriptions();

16 [VB] Public ReadOnly Property ServiceDescriptions As

17 ServiceDescriptionCollection

18 [JScript] public function get ServiceDescriptions() : ServiceDescriptionCollection;

19

20 *Description*

21

22 Style

23 ToString

24

25 [C#] public ServiceDescriptionImportStyle Style {get;}



1 [C++] public: void

2 AddExtensionWarningComments(CodeCommentStatementCollection\* comments,  
3 ServiceDescriptionFormatExtensionCollection\* extensions);

4 [VB] Public Sub AddExtensionWarningComments(ByVal comments As  
5 CodeCommentStatementCollection, ByVal extensions As  
6 ServiceDescriptionFormatExtensionCollection)

7 [JScript] public function AddExtensionWarningComments(comments :  
8 CodeCommentStatementCollection, extensions :  
9 ServiceDescriptionFormatExtensionCollection);

10  
11 *Description*

12  
13 BeginClass

14  
15 [C#] protected abstract CodeTypeDeclaration BeginClass();

16 [C++] protected: virtual CodeTypeDeclaration\* BeginClass() = 0;

17 [VB] MustOverride Protected Function BeginClass() As CodeTypeDeclaration

18 [JScript] protected abstract function BeginClass() : CodeTypeDeclaration;

19  
20 *Description*

21  
22 BeginNamespace

23  
24 [C#] protected virtual void BeginNamespace();

25 [C++] protected: virtual void BeginNamespace();



```
1 || [VB] Overridable Protected Sub BeginNamespace()
```

```
2 || [JScript] protected function BeginNamespace();
```

3

4 *Description*

5

```
6 || EndClass
```

7

```
8  [C#] protected virtual void EndClass();
```

```
9 || [C++] protected: virtual void EndClass();
```

10	[VB] Overridable Protected Sub EndClass()
----	---

```
11 || [JScript] protected function EndClass();
```

12

13 | *Description*

14

```
15 EndNamespace
```

16

```
17  [C#] protected virtual void EndNamespace();
```

```
18  [C++] protected: virtual void EndNamespace();
```

19 | [VB] Overridable Protected Sub EndNamespace()

```
20  [JScript] protected function EndNamespace();
```

21

22 | *Description*

23

24 | GenerateMethod

25

1  
2 [C#] protected abstract CodeMemberMethod GenerateMethod();  
3 [C++] protected: virtual CodeMemberMethod\* GenerateMethod() = 0;  
4 [VB] MustOverride Protected Function GenerateMethod() As  
5 CodeMemberMethod  
6 [JScript] protected abstract function GenerateMethod() : CodeMemberMethod;

7  
8 *Description*

9  
10 **IsBindingSupported**

11  
12 [C#] protected abstract bool IsBindingSupported();  
13 [C++] protected: virtual bool IsBindingSupported() = 0;  
14 [VB] MustOverride Protected Function IsBindingSupported() As Boolean  
15 [JScript] protected abstract function IsBindingSupported() : Boolean;

16  
17 *Description*

18  
19 **IsOperationFlowSupported**

20  
21 [C#] protected abstract bool IsOperationFlowSupported(OperationFlow flow);  
22 [C++] protected: virtual bool IsOperationFlowSupported(OperationFlow flow) =  
23 0;  
24 [VB] MustOverride Protected Function IsOperationFlowSupported(ByVal flow  
25 As OperationFlow) As Boolean

1 [JScript] protected abstract function IsOperationFlowSupported(flow :  
2 OperationFlow) : Boolean;

3  
4 *Description*

5  
6 OperationBindingSyntaxException

7  
8 [C#] public Exception OperationBindingSyntaxException(string text);

9 [C++] public: Exception\* OperationBindingSyntaxException(String\* text);

10 [VB] Public Function OperationBindingSyntaxException(ByVal text As String)

11 As Exception

12 [JScript] public function OperationBindingSyntaxException(text : String) :

13 Exception;

14  
15 *Description*

16  
17 OperationSyntaxException

18  
19 [C#] public Exception OperationSyntaxException(string text);

20 [C++] public: Exception\* OperationSyntaxException(String\* text);

21 [VB] Public Function OperationSyntaxException(ByVal text As String) As

22 Exception

23 [JScript] public function OperationSyntaxException(text : String) : Exception;

24  
25 *Description*

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

## UnsupportedBindingWarning

```
[C#] public void UnsupportedBindingWarning(string text);  
[C++] public: void UnsupportedBindingWarning(String* text);  
[VB] Public Sub UnsupportedBindingWarning(ByVal text As String)  
[JScript] public function UnsupportedBindingWarning(text : String);
```

### *Description*

## UnsupportedOperationBindingWarning

```
[C#] public void UnsupportedOperationBindingWarning(string text);  
[C++] public: void UnsupportedOperationBindingWarning(String* text);  
[VB] Public Sub UnsupportedOperationBindingWarning(ByVal text As String)  
[JScript] public function UnsupportedOperationBindingWarning(text : String);
```

### *Description*

## UnsupportedOperationWarning

```
[C#] public void UnsupportedOperationWarning(string text);  
[C++] public: void UnsupportedOperationWarning(String* text);  
[VB] Public Sub UnsupportedOperationWarning(ByVal text As String)  
[JScript] public function UnsupportedOperationWarning(text : String);
```

1  
2 *Description*

3  
4 ProtocolReflector class (System.Web.Services.Description)  
5 UnsupportedOperationWarning  
6

7  
8 *Description*

9  
10 ProtocolReflector  
11 *Example Syntax:*  
12 UnsupportedOperationWarning  
13

14 [C#] protected ProtocolReflector();  
15 [C++] protected: ProtocolReflector();  
16 [VB] Protected Sub New()  
17 [JScript] protected function ProtocolReflector();

18 Binding  
19 UnsupportedOperationWarning  
20

21 [C#] public Binding Binding {get;}  
22 [C++] public: \_\_property Binding\* get\_Binding();  
23 [VB] Public ReadOnly Property Binding As Binding  
24 [JScript] public function get Binding() : Binding;  
25

*Description*

DefaultNamespace

UnsupportedOperationWarning

[C#] public string DefaultNamespace {get;}

[C++] public: \_\_property String\* get\_DefaultNamespace();

[VB] Public ReadOnly Property DefaultNamespace As String

[JScript] public function get DefaultNamespace() : String;

*Description*

HeaderMessages

UnsupportedOperationWarning

[C#] public MessageCollection HeaderMessages {get;}

[C++] public: \_\_property MessageCollection\* get\_HeaderMessages();

[VB] Public ReadOnly Property HeaderMessages As MessageCollection

[JScript] public function get HeaderMessages() : MessageCollection;

*Description*

InputMessage

UnsupportedOperationWarning

```

1
2 [C#] public Message InputMessage {get;}
3 [C++] public: __property Message* get_InputMessage();
4 [VB] Public ReadOnly Property InputMessage As Message
5 [JScript] public function get InputMessage() : Message;
6

```

### *Description*

Method

UnsupportedOperationWarning

```

12 [C#] public LogicalMethodInfo Method {get;}
13 [C++] public: __property LogicalMethodInfo* get_Method();
14 [VB] Public ReadOnly Property Method As LogicalMethodInfo
15 [JScript] public function get Method() : LogicalMethodInfo;
16

```

### *Description*

MethodAttribute

UnsupportedOperationWarning

```

22 [C#] public WebMethodAttribute MethodAttribute {get;}
23 [C++] public: __property WebMethodAttribute* get_MethodAttribute();
24 [VB] Public ReadOnly Property MethodAttribute As WebMethodAttribute
25 [JScript] public function get MethodAttribute() : WebMethodAttribute;

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

*Description*

Methods  
UnsupportedOperationWarning

```
[C#] public LogicalMethodInfo[] Methods {get;}
[C++] public: __property LogicalMethodInfo* get_Methods();
[VB] Public ReadOnly Property Methods As LogicalMethodInfo ()
[JScript] public function get Methods() : LogicalMethodInfo[];
```

*Description*

Operation  
UnsupportedOperationWarning

```
[C#] public Operation Operation {get;}
[C++] public: __property Operation* get_Operation();
[VB] Public ReadOnly Property Operation As Operation
[JScript] public function get Operation() : Operation;
```

*Description*

OperationBinding  
UnsupportedOperationWarning



```

1
2 [C#] public OperationBinding OperationBinding {get;}
3 [C++] public: __property OperationBinding* get_OperationBinding();
4 [VB] Public ReadOnly Property OperationBinding As OperationBinding
5 [JScript] public function get OperationBinding() : OperationBinding;
6

```

### *Description*

OutputMessage

UnsupportedOperationWarning

```

11
12 [C#] public Message OutputMessage {get;}
13 [C++] public: __property Message* get_OutputMessage();
14 [VB] Public ReadOnly Property OutputMessage As Message
15 [JScript] public function get OutputMessage() : Message;
16

```

### *Description*

Port

UnsupportedOperationWarning

```

21
22 [C#] public Port Port {get;}
23 [C++] public: __property Port* get_Port();
24 [VB] Public ReadOnly Property Port As Port
25 [JScript] public function get Port() : Port;

```

1  
2 *Description*

3  
4 PortType

5 UnsupportedOperationWarning

6  
7 [C#] public PortType PortType {get;}

8 [C++] public: \_\_property PortType\* get\_PortType();

9 [VB] Public ReadOnly Property PortType As PortType

10 [JScript] public function get PortType() : PortType;

11  
12 *Description*

13  
14 ProtocolName

15 UnsupportedOperationWarning

16  
17 [C#] public abstract string ProtocolName {get;}

18 [C++] public: \_\_property virtual String\* get\_ProtocolName() = 0;

19 [VB] MustOverride Public ReadOnly Property ProtocolName As String

20 [JScript] public abstract function get ProtocolName() : String;

21  
22 *Description*

23  
24 ReflectionImporter

25 UnsupportedOperationWarning

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#] public XmlReflectionImporter ReflectionImporter {get;}
[C++] public: __property XmlReflectionImporter* get_ReflectionImporter();
[VB] Public ReadOnly Property ReflectionImporter As XmlReflectionImporter
[JScript] public function get ReflectionImporter() : XmlReflectionImporter;
```

*Description*

SchemaExporter  
UnsupportedOperationWarning

```
[C#] public XmlSchemaExporter SchemaExporter {get;}
[C++] public: __property XmlSchemaExporter* get_SchemaExporter();
[VB] Public ReadOnly Property SchemaExporter As XmlSchemaExporter
[JScript] public function get SchemaExporter() : XmlSchemaExporter;
```

*Description*

Schemas  
UnsupportedOperationWarning

```
[C#] public XmlSchemas Schemas {get;}
[C++] public: __property XmlSchemas* get_Schemas();
[VB] Public ReadOnly Property Schemas As XmlSchemas
[JScript] public function get Schemas() : XmlSchemas;
```

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25

*Description*

Service

## UnsupportedOperationWarning

```
[C#] public Service Service {get;}
```

```
[C++] public: __property Service* get_Service();
```

[VB] Public ReadOnly Property Service As Service

```
[JScript] public function get Service() : Service;
```

### Description

## ServiceDescription

## UnsupportedOperationWarning

```
[C#] public ServiceDescription ServiceDescription {get;}
```

```
[C++] public: __property ServiceDescription* get_ServiceDescription();
```

[VB] Public ReadOnly Property ServiceDescription As ServiceDescription

```
[JScript] public function get ServiceDescription() : ServiceDescription;
```

*Description*

## ServiceDescriptions

## UnsupportedOperationWarning

```

1
2 [C#] public ServiceDescriptionCollection ServiceDescriptions {get;}
3 [C++] public: __property ServiceDescriptionCollection*
4 get_ServiceDescriptions();
5 [VB] Public ReadOnly Property ServiceDescriptions As
6 ServiceDescriptionCollection
7 [JScript] public function get ServiceDescriptions() : ServiceDescriptionCollection;
8
9 Description
10
11     ServiceType
12     UnsupportedOperationWarning
13
14 [C#] public Type ServiceType {get;}
15 [C++] public: __property Type* get_ServiceType();
16 [VB] Public ReadOnly Property ServiceType As Type
17 [JScript] public function get ServiceType() : Type;
18
19 Description
20
21     ServiceUrl
22     UnsupportedOperationWarning
23
24 [C#] public string ServiceUrl {get;}
25 [C++] public: __property String* get_ServiceUrl();

```

1 [VB] Public ReadOnly Property ServiceUrl As String

2 [JScript] public function get ServiceUrl() : String;

3  
4 *Description*

5  
6 BeginClass

7  
8 [C#] protected virtual void BeginClass();

9 [C++] protected: virtual void BeginClass();

10 [VB] Overridable Protected Sub BeginClass()

11 [JScript] protected function BeginClass();

12  
13 *Description*

14  
15 EndClass

16  
17 [C#] protected virtual void EndClass();

18 [C++] protected: virtual void EndClass();

19 [VB] Overridable Protected Sub EndClass()

20 [JScript] protected function EndClass();

21  
22 *Description*

23  
24 GetServiceDescription

FOI b7D b7C b6 b7C b7D

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#] public ServiceDescription GetServiceDescription(string ns);
[C++] public: ServiceDescription* GetServiceDescription(String* ns);
[VB] Public Function GetServiceDescription(ByVal ns As String) As
ServiceDescription
[JavaScript] public function GetServiceDescription(ns : String) : ServiceDescription;
```

*Description*

ReflectMethod

```
[C#] protected abstract bool ReflectMethod();
[C++] protected: virtual bool ReflectMethod() = 0;
[VB] MustOverride Protected Function ReflectMethod() As Boolean
[JavaScript] protected abstract function ReflectMethod() : Boolean;
```

*Description*

ReflectMethodBinding

```
[C#] protected virtual string ReflectMethodBinding();
[C++] protected: virtual String* ReflectMethodBinding();
[VB] Overridable Protected Function ReflectMethodBinding() As String
[JavaScript] protected function ReflectMethodBinding() : String;
```

*Description*

Service class (System.Web.Services.Description)

ToString

*Description*

Groups together a set of related **System.Web.Services.Description.Port** instances associated with a Web Service. This class cannot be inherited.

Service

*Example Syntax:*

ToString

[C#] public Service();

[C++] public: Service();

[VB] Public Sub New()

[JScript] public function Service();

Documentation

Extensions

ToString

*Description*



1 Gets the collection of extensibility elements contained in the  
2 **System.Web.Services.Description.Service** instance.

3 This property will generally only be populated with members if the  
4 members of the **System.Web.Services.Description.Service.Ports** property lack  
5 extensibility elements.

6 Name

7 ToString

8  
9 [C#] public string Name {get; set;}

10 [C++] public: \_\_property String\* get\_Name();public: \_\_property void  
11 set\_Name(String\*);

12 [VB] Public Property Name As String

13 [JScript] public function get Name() : String;public function set Name(String);

14  
15 *Description*

16 Gets or sets the name of the **System.Web.Services.Description.Service**  
17 instance.

18 Ports

19 ToString

20  
21 [C#] public PortCollection Ports {get;}

22 [C++] public: \_\_property PortCollection\* get\_Ports();

23 [VB] Public ReadOnly Property Ports As PortCollection

24 [JScript] public function get Ports() : PortCollection;



1 ToString  
2 **System.Web.Services.Description.Service**

3  
4 *Description*

5 Gets or sets the value of a **System.Web.Services.Description.Service** at  
6 the specified zero-based index. The zero-based index of the **Service** to be  
7 modified or returned.

8 Item

9 ToString

10  
11 [C#] public Service this[string name] {get;}

12 [C++] public: \_\_property Service\* get\_Item(String\* name);

13 [VB] Public Default ReadOnly Property Item(ByVal name As String) As Service

14 [JScript] returnValue = ServiceCollectionObject.Item(name);

15  
16 *Description*

17 Gets the **Service** instance by the value of its  
18 **System.Web.Services.Description.Service.Name** property. The string value  
19 representing the name of the **System.Web.Services.Description.Service** returned.

20 List

21 Table

22 Add

23  
24 [C#] public int Add(Service service);

25 [C++] public: int Add(Service\* service);

1 [VB] Public Function Add(ByVal service As Service) As Integer

2 [JScript] public function Add(service : Service) : int;

3  
4 *Description*

5 Adds the specified **System.Web.Services.Description.Service** instance to  
6 the end of the **ServiceCollection** .

7 *Return Value:* Returns the zero-based index where *service* has been added. The  
8 **System.Web.Services.Description.Service** instance to be added to the  
9 **ServiceCollection**.

10 Contains

11  
12 [C#] public bool Contains(Service service);

13 [C++] public: bool Contains(Service\* service);

14 [VB] Public Function Contains(ByVal service As Service) As Boolean

15 [JScript] public function Contains(service : Service) : Boolean;

16  
17 *Description*

18 Gets a value indicating whether the specified  
19 **System.Web.Services.Description.Service** instance is a member of the  
20 **ServiceCollection** .

21 *Return Value:* **true** if the specified **System.Web.Services.Description.Service**  
22 instance is a member of the **ServiceCollection** ; otherwise, **false** . A  
23 **System.Web.Services.Description.Service** object.

24 CopyTo

```

1
2 [C#] public void CopyTo(Service[] array, int index);
3 [C++] public: void CopyTo(Service* array[], int index);
4 [VB] Public Sub CopyTo(ByVal array() As Service, ByVal index As Integer)
5 [JScript] public function CopyTo(array : Service[], index : int);
6

```

### *Description*

Copies the entire **ServiceCollection** to a one-dimensional array of type **System.Web.Services.Description.Service**, starting at the specified zero-based index of the target array. An array of type **System.Web.Services.Description.Service** serving as the destination of the copy action. The zero-based index at which to start placing the copied **ServiceCollection**.

### *GetKey*

```

14
15
16 [C#] protected override string GetKey(object value);
17 [C++] protected: String* GetKey(Object* value);
18 [VB] Overrides Protected Function GetKey(ByVal value As Object) As String
19 [JScript] protected override function GetKey(value : Object) : String;
20

```

### *Description*

Returns the name of the **System.Web.Services.Description.Service** associated with the value passed by reference. An object for which to return the name.

### *IndexOf*

```

1
2 [C#] public int IndexOf(Service service);
3 [C++] public: int IndexOf(Service* service);
4 [VB] Public Function IndexOf(ByVal service As Service) As Integer
5 [JScript] public function IndexOf(service : Service) : int;
6

```

### *Description*

Searches for the specified **System.Web.Services.Description.Service** instance and returns the zero-based index of the first occurrence within the **ServiceCollection**.

*Return Value:* Returns a 32-bit signed integer. A **System.Web.Services.Description.Service** object.

### *Insert*

```

15 [C#] public void Insert(int index, Service service);
16 [C++] public: void Insert(int index, Service* service);
17 [VB] Public Sub Insert(ByVal index As Integer, ByVal service As Service)
18 [JScript] public function Insert(index : int, service : Service);
19

```

### *Description*

Adds the specified **System.Web.Services.Description.Service** instance to the **ServiceCollection** at the specified zero-based index.

If the number of items in the collection already equals the collection's capacity, the capacity is doubled by automatically reallocating the internal array before the new element is inserted. The zero-based index at which to insert the

specified **System.Web.Services.Description.Service**. The  
**System.Web.Services.Description.Service** instance to be added to the  
**ServiceCollection**.

#### Remove

[C#] public void Remove(Service service);  
[C++] public: void Remove(Service\* service);  
[VB] Public Sub Remove(ByVal service As Service)  
[JScript] public function Remove(service : Service);

#### *Description*

Removes the first occurrence of the specified  
**System.Web.Services.Description.Service** from the **ServiceCollection** .

This method performs a linear search; therefore, the average execution time  
is proportional to **System.Web.Services.Description.ServiceCollection.Count** .

A **System.Web.Services.Description.Service** object.

#### SetParent

[C#] protected override void SetParent(object value, object parent);  
[C++] protected: void SetParent(Object\* value, Object\* parent);  
[VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As  
Object)  
[JScript] protected override function SetParent(value : Object, parent : Object);

#### *Description*

1       Sets the parent **System.Web.Services.Description.ServiceDescription** of  
2 a member of **System.Web.Services.Description.ServiceCollection** . An object,  
3 of type **System.Web.Services.Description.Service**, within the **ServiceCollection**.  
4 The object, of type **System.Web.Services.Description.ServiceDescription**, to be  
5 set as the parent.

6       ServiceDescription class (System.Web.Services.Description)

7       ToString

8  
9  
10    *Description*

11       Provides a means of creating and formatting a valid WSDL (XML)  
12 document file, complete with appropriate namespaces, elements and attributes, for  
13 describing a Web Service. This class cannot be inherited.

14       Instances of this class can be created with either the **new** keyword or the  
15 **static**

16 **System.Web.Services.Description.ServiceDescription.Read(System.IO.TextRe**  
17 **ader)** method, which parses a WSDL (Web Service Description Language) file  
18 and assigns its values to appropriate members of the class.

19       ToString

20  
21    [C#] public const string Namespace;

22    [C++] public: const String\* Namespace;

23    [VB] Public Const Namespace As String

24    [JScript] public var Namespace : String;



## Description

The XML namespace in which the **ServiceDescription** class is defined ("http://schemas.xmlsoap.org/wsdl/"). This field is constant.

ServiceDescription

*Example Syntax:*

ToString

```
[C#] public ServiceDescription();
```

```
[C++] public: ServiceDescription();
```

```
[VB] Public Sub New()
```

```
[JScript] public function ServiceDescription();
```

Bindings

ToString

```
[C#] public BindingCollection Bindings {get;}
```

```
[C++] public: __property BindingCollection* get_Bindings();
```

```
[VB] Public ReadOnly Property Bindings As BindingCollection
```

```
[JScript] public function get Bindings() : BindingCollection;
```

## Description

Represents the collection of **System.Web.Services.Description.Binding** elements that the **System.Web.Services.Description.ServiceDescription** contains.

Documentation

Extensions

ToString

*Description*

Gets the collection of extensibility elements that the **System.Web.Services.Description.ServiceDescription** contains.

Imports

ToString

[C#] public ImportCollection Imports {get;}

[C++] public: \_\_property ImportCollection\* get\_Imports();

[VB] Public ReadOnly Property Imports As ImportCollection

[JScript] public function get Imports() : ImportCollection;

*Description*

Gets the collection of **System.Web.Services.Description.Import** elements that the **System.Web.Services.Description.ServiceDescription** contains.

Messages

ToString

[C#] public MessageCollection Messages {get;}

[C++] public: \_\_property MessageCollection\* get\_Messages();

[VB] Public ReadOnly Property Messages As MessageCollection

[JScript] public function get Messages() : MessageCollection;

1  
2 *Description*

3 The collection of **System.Web.Services.Description.Message** elements  
4 the **System.Web.Services.Description.ServiceDescription** contains.

5 Name

6 ToString

7  
8 [C#] public string Name {get; set;}

9 [C++] public: \_\_property String\* get\_Name();public: \_\_property void  
10 set\_Name(String\*);

11 [VB] Public Property Name As String

12 [JScript] public function get Name() : String;public function set Name(String);

13  
14 *Description*

15 Gets or sets the XML **Name** attribute of the **Descriptions** tag enclosing the  
16 WSDL file.

17 Returns an empty string ("" ) if this property value has not been assigned.

18 PortTypes

19 ToString

20  
21 [C#] public PortTypeCollection PortTypes {get;}

22 [C++] public: \_\_property PortTypeCollection\* get\_PortTypes();

23 [VB] Public ReadOnly Property PortTypes As PortTypeCollection

24 [JScript] public function get PortTypes() : PortTypeCollection;

## Description

The collection of **System.Web.Services.Description.PortType** elements the **System.Web.Services.Description.ServiceDescription** contains.

RetrievalUrl

ToString

[C#] public string RetrievalUrl {get; set;}

[C++] public: \_\_property String\* get\_RetrievalUrl();public: \_\_property void set\_RetrievalUrl(String\*);

[VB] Public Property RetrievalUrl As String

[JScript] public function get RetrievalUrl() : String;public function set RetrievalUrl(String);

Serializer

ToString

[C#] public static XmlSerializer Serializer {get;}

[C++] public: \_\_property static XmlSerializer\* get\_Serializer();

[VB] Public Shared ReadOnly Property Serializer As XmlSerializer

[JScript] public static function get Serializer() : XmlSerializer; Returns the serializer for processing web service calls. The serializer is customized according to settings in config.web.

ServiceDescriptions

ToString

```

1
2 [C#] public ServiceDescriptionCollection ServiceDescriptions {get;}
3 [C++] public: __property ServiceDescriptionCollection*
4 get_ServiceDescriptions();
5 [VB] Public ReadOnly Property ServiceDescriptions As
6 ServiceDescriptionCollection
7 [JScript] public function get ServiceDescriptions() : ServiceDescriptionCollection;
8

```

### *Description*

The **System.Web.Services.Description.ServiceDescriptionCollection** instance of which the **System.Web.Services.Description.ServiceDescription** is a member.

This property is read-only.

Services

ToString

```

17 [C#] public ServiceCollection Services {get;}
18 [C++] public: __property ServiceCollection* get_Services();
19 [VB] Public ReadOnly Property Services As ServiceCollection
20 [JScript] public function get Services() : ServiceCollection;
21

```

### *Description*

Gets the collection of **System.Web.Services.Description.Service** instances that the **System.Web.Services.Description.ServiceDescription** contains.

TargetNamespace

ToString

[C#] public string TargetNamespace {get; set;} ^

[C++] public: \_\_property String\* get\_TargetNamespace();public: \_\_property void  
set\_TargetNamespace(String\*);

[VB] Public Property TargetNamespace As String

[JScript] public function get TargetNamespace() : String;public function set  
TargetNamespace(String);

#### *Description*

Gets or sets the XML **targetNamespace** attribute of the **Descriptions** tag  
enclosing a WSDL file.

Types

ToString

[C#] public Types Types {get; set;}

[C++] public: \_\_property Types\* get\_Types();public: \_\_property void  
set\_Types(Types\*);

[VB] Public Property Types As Types

[JScript] public function get Types() : Types;public function set Types(Types);

#### *Description*

Gets or sets the **System.Web.Services.Description.Types** contained by the  
**ServiceDescription** instance.

CanRead

```

1
2 [C#] public static bool CanRead(XmlReader reader);
3 [C++] public: static bool CanRead(XmlReader* reader);
4 [VB] Public Shared Function CanRead(ByVal reader As XmlReader) As Boolean
5 [JScript] public static function CanRead(reader : XmlReader) : Boolean;
6

```

### *Description*

Gets a value indicating whether an **System.Xml.XmlReader** represents a valid WSDL file that can be parsed.

*Return Value:* **True** if the **System.Xml.Serialization.XmlSerializer** can recognize the node on which the **System.Xml.XmlReader** is positioned; otherwise **false** . An **System.Xml.XmlReader**

### *Read*

```

13
14
15 [C#] public static ServiceDescription Read(Stream stream);
16 [C++] public: static ServiceDescription* Read(Stream* stream);
17 [VB] Public Shared Function Read(ByVal stream As Stream) As
18 ServiceDescription
19 [JScript] public static function Read(stream : Stream) : ServiceDescription;
20

```

### *Description*

Initializes an instance of a **System.Web.Services.Description.ServiceDescription** object by directly loading the XML from a **System.IO.Stream** instance.

*Return Value:* Returns a **System.Web.Services.Description.ServiceDescription**

object. A **System.IO.Stream** object, passed by reference, that contains the bytes to be read in.

### Read

[C#] public static ServiceDescription Read(string fileName);

[C++] public: static ServiceDescription\* Read(String\* fileName);

[VB] Public Shared Function Read(ByVal fileName As String) As

ServiceDescription

[JScript] public static function Read(fileName : String) : ServiceDescription;

### *Description*

Initializes an instance of a **System.Web.Services.Description.ServiceDescription** object by directly loading the XML from the specified file.

*Return Value:* Returns a **System.Web.Services.Description.ServiceDescription** object. A string representing the path to the file to be read in.

### Read

[C#] public static ServiceDescription Read(TextReader textReader);

[C++] public: static ServiceDescription\* Read(TextReader\* textReader);

[VB] Public Shared Function Read(ByVal textReader As TextReader) As

ServiceDescription

[JScript] public static function Read(textReader : TextReader) :

ServiceDescription; Initializes an instance of a

**System.Web.Services.Description.ServiceDescription** object by directly loading



the XML.

### *Description*

Initializes an instance of a **System.Web.Services.Description.ServiceDescription** object by directly loading the XML from a **System.IO.TextReader** instance.

*Return Value:* Returns a **System.Web.Services.Description.ServiceDescription** object. A **System.IO.TextReader** object, passed by reference, that contains the text to be read in.

### *Read*

[C#] public static ServiceDescription Read(XmlReader reader);

[C++] public: static ServiceDescription\* Read(XmlReader\* reader);

[VB] Public Shared Function Read(ByVal reader As XmlReader) As ServiceDescription

[JScript] public static function Read(reader : XmlReader) : ServiceDescription;

### *Description*

Initializes an instance of a **System.Web.Services.Description.ServiceDescription** object by directly loading the XML from an **System.Xml.XmlReader** instance.

*Return Value:* Returns a **System.Web.Services.Description.ServiceDescription** object. An **System.Xml.XmlReader** instance, passed by reference, that contains the XML data to be read in.

### *Write*

```

1
2 [C#] public void Write(Stream stream);
3 [C++] public: void Write(Stream* stream);
4 [VB] Public Sub Write(ByVal stream As Stream)
5 [JScript] public function Write(stream : Stream);
6

```

#### *Description*

Writes out the **System.Web.Services.Description.ServiceDescription** to the specified **System.IO.Stream** instance. A **System.IO.Stream**, passed by reference, that contains the WSDL file produced.

#### *Write*

```

11
12
13 [C#] public void Write(string fileName);
14 [C++] public: void Write(String* fileName);
15 [VB] Public Sub Write(ByVal fileName As String)
16 [JScript] public function Write(fileName : String); Writes out the
17 System.Web.Services.Description.ServiceDescription as a WSDL file.
18

```

#### *Description*

Writes out the **System.Web.Services.Description.ServiceDescription** as a WSDL file to the specified path. A string representing the physical path to which the WSDL file is written.

#### *Write*

```

23
24
25 [C#] public void Write(TextWriter writer);

```

1 [C++] public: void Write(TextWriter\* writer);

2 [VB] Public Sub Write(ByVal writer As TextWriter)

3 [JScript] public function Write(writer : TextWriter);

4  
5 *Description*

6 Writes out the **System.Web.Services.Description.ServiceDescription** to  
7 the **System.IO.TextWriter** instance. A **System.IO.TextWriter** instance.

8 Write

9  
10 [C#] public void Write(XmlWriter writer);

11 [C++] public: void Write(XmlWriter\* writer);

12 [VB] Public Sub Write(ByVal writer As XmlWriter)

13 [JScript] public function Write(writer : XmlWriter);

14  
15 *Description*

16 Writes out the **System.Web.Services.Description.ServiceDescription** to  
17 the **System.Xml.XmlWriter** as a WSDL file. An **System.Xml.XmlWriter**,  
18 passed by reference, that contains the WSDL file produced.

19 ServiceDescriptionBaseCollection class  
20 (System.Web.Services.Description)

21 Write

22  
23  
24 *Description*

25

1 Forms the basis for the strongly typed collections that are members of the  
2 **System.Web.Services.Description** namespace.

3 Count

4 InnerList

5 List

6 Table

7 Write

8  
9  
10 *Description*

11 Gets an interface that implements the association of the keys and values in  
12 the **System.Web.Services.Description.ServiceDescriptionBaseCollection**  
13 instance.

14 The default implementation returns a **System.Collections.Hashtable** to  
15 associate the keys and values. This protected method is accessible only through  
16 this class or a derived class.

17 GetKey

18  
19 [C#] protected virtual string GetKey(object value);

20 [C++] protected: virtual String\* GetKey(Object\* value);

21 [VB] Overridable Protected Function GetKey(ByVal value As Object) As String

22 [JScript] protected function GetKey(value : Object) : String;

23  
24 *Description*

25 Returns the name of the key associated with the value passed by reference.

1 The default implementation of this method is intended to be overridden by  
2 a derived class to return the name of the key associated with *value* . Note that the  
3 default implementation returns null ( **Nothing** in Visual Basic). This protected  
4 method is accessible only through this class or a derived class. An object for  
5 which to return the name of the key.

#### 6 OnClear

7  
8 [C#] protected override void OnClear();

9 [C++] protected: void OnClear();

10 [VB] Overrides Protected Sub OnClear()

11 [JScript] protected override function OnClear();

#### 12 13 *Description*

14 Clears the contents of the  
15 **System.Web.Services.Description.ServiceDescriptionBaseCollection** instance.

16 This method is intended to be overridden to perform additional processes  
17 when clearing the contents of a derived Collection. This protected method is  
18 accessible only through this class or a derived class.

#### 19 OnInsertComplete

20  
21 [C#] protected override void OnInsertComplete(int index, object value);

22 [C++] protected: void OnInsertComplete(int index, Object\* value);

23 [VB] Overrides Protected Sub OnInsertComplete(ByVal index As Integer, ByVal  
24 value As Object)

25 [JScript] protected override function OnInsertComplete(index : int, value :

Object);

### *Description*

Performs additional custom processes when inserting a new element into the **System.Web.Services.Description.ServiceDescriptionBaseCollection** instance.

The default implementation of this method is intended to be overridden by a derived class to perform some action when the specified element is inserted. The zero-based index at which to insert the object represented by the *value* parameter.

The new value of the element at the specified zero-based index.

### **OnRemove**

[C#] protected override void OnRemove(int index, object value);

[C++] protected: void OnRemove(int index, Object\* value);

[VB] Overrides Protected Sub OnRemove(ByVal index As Integer, ByVal value As Object)

[JScript] protected override function OnRemove(index : int, value : Object);

### *Description*

Removes an element from the **System.Web.Services.Description.ServiceDescriptionBaseCollection** instance.

This method is intended to be overridden to perform additional processes when an element is removed from the **ServiceDescriptionBaseCollection** instance. The zero-based index at which the object represented by the *value* parameter can be found. The object to remove from the collection.

## OnSet

[C#] protected override void OnSet(int index, object oldValue, object newValue);  
[C++] protected: void OnSet(int index, Object\* oldValue, Object\* newValue);  
[VB] Overrides Protected Sub OnSet(ByVal index As Integer, ByVal oldValue As Object, ByVal newValue As Object)  
[JScript] protected override function OnSet(index : int, oldValue : Object, newValue : Object);

### *Description*

Replaces one value with another within the **System.Web.Services.Description.ServiceDescriptionBaseCollection** instance.

The default implementation of this method is intended to be overridden to perform additional processes when setting the value of an element in the **ServiceDescriptionBaseCollection** instance. This protected method is accessible only through this class or a derived class. The zero-based index where the object represented by the *oldValue* parameter can be found. The object to replace with the object represented by the *newValue* parameter. The object that replaces the object represented by the *oldValue* parameter.

## SetParent

[C#] protected virtual void SetParent(object value, object parent);  
[C++] protected: virtual void SetParent(Object\* value, Object\* parent);  
[VB] Overridable Protected Sub SetParent(ByVal value As Object, ByVal parent As Object)

1 [JScript] protected function SetParent(value : Object, parent : Object);

2  
3 *Description*

4       Sets the Parent property of the  
5 **System.Web.Services.Description.ServiceDescriptionBaseCollection** instance.

6       The default implementation of this method is intended to be overridden by  
7 a derived class to set a parent of an appropriate class. Note also that the default  
8 implementation performs no action. This protected method is accessible only  
9 through this class or a derived class. The object for which to set the parent object.

10 The object to set as the parent.

11       ServiceDescriptionCollection class (System.Web.Services.Description)

12       ToString

13  
14  
15 *Description*

16       Represents a collection of  
17 **System.Web.Services.Description.ServiceDescription** instances. This class  
18 cannot be inherited.

19       ServiceDescriptionCollection

20       *Example Syntax:*

21       ToString

22  
23 [C#] public ServiceDescriptionCollection();

24 [C++] public: ServiceDescriptionCollection();

25 [VB] Public Sub New()



1 [JScript] public function ServiceDescriptionCollection();

2  
3 *Description*

4       Initializes a new instance of the  
5 **System.Web.Services.Description.ServiceDescriptionCollection** class.

6       Count

7       InnerList

8       Item

9       ToString

10      **System.Web.Services.Description.ServiceDescription**

11  
12 *Description*

13       Gets or sets the value of a  
14 **System.Web.Services.Description.ServiceDescription** at the specified zero-  
15 based index. The zero-based index of the  
16 **System.Web.Services.Description.ServiceDescription** whose value is modified  
17 or returned.

18       Item

19       ToString

20  
21 [C#] public ServiceDescription this[string ns] {get;}

22 [C++] public: \_\_property ServiceDescription\* get\_Item(String\* ns);

23 [VB] Public Default ReadOnly Property Item(ByVal ns As String) As  
24 ServiceDescription

25 [JScript] returnValue = ServiceDescriptionCollectionObject.Item(ns);

## *Description*

Gets a **System.Web.Services.Description.ServiceDescription** specified by its **System.Web.Services.Description.ServiceDescription.TargetNamespace** property. The namespace of the **System.Web.Services.Description.ServiceDescription** to be returned.

List

Table

Add

[C#] public int Add(ServiceDescription serviceDescription);

[C++] public: int Add(ServiceDescription\* serviceDescription);

[VB] Public Function Add(ByVal serviceDescription As ServiceDescription) As Integer

[JScript] public function Add(serviceDescription : ServiceDescription) : int;

## *Description*

Adds the specified **System.Web.Services.Description.ServiceDescription** to the end of the **ServiceDescriptionCollection**.

*Return Value:* Returns the zero-based index where the specified

**System.Web.Services.Description.ServiceDescription** instance has been added.

The **System.Web.Services.Description.ServiceDescription** to be added to the collection.

Contains

```

1
2 [C#] public bool Contains(ServiceDescription serviceDescription);
3 [C++] public: bool Contains(ServiceDescription* serviceDescription);
4 [VB] Public Function Contains(ByVal serviceDescription As ServiceDescription)
5 As Boolean
6 [JScript] public function Contains(serviceDescription : ServiceDescription) :
7 Boolean;
8

```

### *Description*

Gets a value indicating whether the specified **System.Web.Services.Description.ServiceDescription** instance is a member of the collection.

*Return Value:* **true** if the specified **ServiceDescription** instance is a member of the **ServiceDescriptionCollection** ; otherwise **false** . A **System.Web.Services.Description.ServiceDescription** object.

### *CopyTo*

```

18 [C#] public void CopyTo(ServiceDescription[] array, int index);
19 [C++] public: void CopyTo(ServiceDescription* array[], int index);
20 [VB] Public Sub CopyTo(ByVal array() As ServiceDescription, ByVal index As
21 Integer)
22 [JScript] public function CopyTo(array : ServiceDescription[], index : int);
23

```

### *Description*

Copies the entire **ServiceDescriptionCollection** to a one-dimensional array of type **System.Web.Services.Description.ServiceDescription** , starting at the specified zero-based index of the target array. An array of type **System.Web.Services.Description.ServiceDescription** serving as the destination of the copy action. The zero-based index at which to start placing the copied collection.

#### GetBinding

[C#] public Binding GetBinding(XmlQualifiedName name);

[C++] public: Binding\* GetBinding(XmlQualifiedName\* name);

[VB] Public Function GetBinding(ByVal name As XmlQualifiedName) As

Binding

[JScript] public function GetBinding(name : XmlQualifiedName) : Binding;

#### Description

Searches the **ServiceDescriptionCollection** and returns the **System.Web.Services.Description.Binding** instance with the specified name that is a member of one of the **System.Web.Services.Description.ServiceDescription** instances contained in the collection.

*Return Value:* A Binding object. The **System.Xml.XmlQualifiedName**, passed by reference, whose **Name** property is shared by the

**System.Web.Services.Description.Binding** returned.

#### GetKey

[C#] protected override string GetKey(object value);

1 [C++] protected: String\* GetKey(Object\* value);

2 [VB] Overrides Protected Function GetKey(ByVal value As Object) As String

3 [JScript] protected override function GetKey(value : Object) : String;

4  
5 *Description*

6 Returns the

7 **System.Web.Services.Description.ServiceDescription.TargetNamespace**

8 property of the **System.Web.Services.Description.ServiceDescription** associated

9 with the value passed by reference. The object for which to return the

10 TargetNamespace.

11 GetMessage

12  
13 [C#] public Message GetMessage(XmlQualifiedName name);

14 [C++] public: Message\* GetMessage(XmlQualifiedName\* name);

15 [VB] Public Function GetMessage(ByVal name As XmlQualifiedName) As

16 Message

17 [JScript] public function GetMessage(name : XmlQualifiedName) : Message;

18  
19 *Description*

20 Searches the **ServiceDescriptionCollection** and returns the

21 **System.Web.Services.Description.Message** instance with the specified name that

22 is a member of one of the **System.Web.Services.Description.ServiceDescription**

23 instances contained in the collection.

24 *Return Value:* A **System.Web.Services.Description.Message** object. The

1 **System.Xml.XmlQualifiedName**, passed by reference, whose **Name** property is  
2 shared by the **System.Web.Services.Description.Message** returned.

### 3 GetPortType

4  
5 [C#] public PortType GetPortType(XmlQualifiedName name);  
6 [C++] public: PortType\* GetPortType(XmlQualifiedName\* name);  
7 [VB] Public Function GetPortType(ByVal name As XmlQualifiedName) As  
8 PortType  
9 [JScript] public function GetPortType(name : XmlQualifiedName) : PortType;

### 11 *Description*

12 Searches the **ServiceDescriptionCollection** and returns the  
13 **System.Web.Services.Description.PortType** instance with the specified name  
14 that is a member of one of the  
15 **System.Web.Services.Description.ServiceDescription** instances contained in the  
16 collection.

17 *Return Value:* A **System.Web.Services.Description.PortType** object. The  
18 **System.Xml.XmlQualifiedName**, passed by reference, whose **Name** property is  
19 shared by the **System.Web.Services.Description.PortType** returned.

### 20 GetService

21  
22 [C#] public Service GetService(XmlQualifiedName name);  
23 [C++] public: Service\* GetService(XmlQualifiedName\* name);  
24 [VB] Public Function GetService(ByVal name As XmlQualifiedName) As Service  
25 [JScript] public function GetService(name : XmlQualifiedName) : Service;

## Description

Searches the **ServiceDescriptionCollection** and returns the **System.Web.Services.Description.Service** instance with the specified name that is a member of one of the **System.Web.Services.Description.ServiceDescription** instances contained in the collection.

*Return Value:* A **System.Web.Services.Description.Service** object. The **System.Xml.XmlQualifiedName**, passed by reference, whose **Name** property is shared by the **System.Web.Services.Description.Service** returned.

## IndexOf

[C#] public int IndexOf(ServiceDescription serviceDescription);  
[C++] public: int IndexOf(ServiceDescription\* serviceDescription);  
[VB] Public Function IndexOf(ByVal serviceDescription As ServiceDescription)  
As Integer  
[JScript] public function IndexOf(serviceDescription : ServiceDescription) : int;

## Description

Searches for the specified **System.Web.Services.Description.ServiceDescription** instance and returns the zero-based index of the first occurrence within the **ServiceDescriptionCollection**.

*Return Value:* Returns a 32-bit signed integer. A **System.Web.Services.Description.ServiceDescription** object.

## Insert

```

1  [C#] public void Insert(int index, ServiceDescription serviceDescription);
2
3  [C++] public: void Insert(int index, ServiceDescription* serviceDescription);
4
5  [VB] Public Sub Insert(ByVal index As Integer, ByVal serviceDescription As
6  ServiceDescription)
7
8  [JScript] public function Insert(index : int, serviceDescription :
9  ServiceDescription);

```

### *Description*

Adds the specified **System.Web.Services.Description.ServiceDescription** instance to the **ServiceDescriptionCollection** at the specified index. The zero-based index at which to insert the specified **System.Web.Services.Description.ServiceDescription**. The **System.Web.Services.Description.ServiceDescription** to add to the collection.

### *Remove*

```

17 [C#] public void Remove(ServiceDescription serviceDescription);
18
19 [C++] public: void Remove(ServiceDescription* serviceDescription);
20
21 [VB] Public Sub Remove(ByVal serviceDescription As ServiceDescription)
22
23 [JScript] public function Remove(serviceDescription : ServiceDescription);

```

### *Description*

Removes the first occurrence of a **System.Web.Services.Description.ServiceDescription** instance specified by the *serviceDescription* parameter from the **ServiceDescriptionCollection**.



This method performs a linear search; therefore, the average execution time is proportional to **Count** . The **System.Web.Services.Description.ServiceDescription** to be removed from the collection.

ServiceDescriptionFormatExtension class  
(System.Web.Services.Description)

ToString

### *Description*

Represents an extensibility element added to a Web Service.

In a derived class, the **ServiceDescriptionFormatExtension** class allows users to define extensibility elements in addition to those defined in the WSDL specification. Note that extensibility elements can be added to a **ServiceDescription** at any of several levels. Thus multiple classes derived from the **System.Web.Services.Description.DocumentableItem** class (including the **System.Web.Services.Description.ServiceDescription** class) have an **Extensions** property, which returns a **System.Web.Services.Description.ServiceDescriptionFormatExtensionCollection** .

ServiceDescriptionFormatExtension

*Example Syntax:*

ToString

[C#] protected ServiceDescriptionFormatExtension();

1 [C++] protected: ServiceDescriptionFormatExtension();  
2 [VB] Protected Sub New()  
3 [JScript] protected function ServiceDescriptionFormatExtension();  
4       Handled  
5       ToString  
6  
7 [C#] public bool Handled {get; set;}  
8 [C++] public: \_\_property bool get\_Handled();public: \_\_property void  
9 set\_Handled(bool);  
10 [VB] Public Property Handled As Boolean  
11 [JScript] public function get Handled() : Boolean;public function set  
12 Handled(Boolean);  
13

#### *Description*

15       Gets or sets a value indicating whether the  
16 **System.Web.Services.Description.ServiceDescriptionFormatExtension** is  
17 handled by the action to which it refers.

18       Parent

19       ToString  
20

21 [C#] public object Parent {get;}  
22 [C++] public: \_\_property Object\* get\_Parent();  
23 [VB] Public ReadOnly Property Parent As Object  
24 [JScript] public function get Parent() : Object;  
25

1  
2 *Description*

3 Gets the parent object of the

4 **System.Web.Services.Description.ServiceDescriptionFormatExtension** .

5 When overridden in a derived class, the class returned by this property will  
6 be a derived class rather than the base **System.Object** class.

7 Required

8 ToString

9  
10 [C#] public bool Required {get; set;}

11 [C++] public: \_\_property bool get\_Required();public: \_\_property void  
12 set\_Required(bool);

13 [VB] Public Property Required As Boolean

14 [JScript] public function get Required() : Boolean;public function set  
15 Required(Boolean);

16  
17 *Description*

18 Gets or sets a value indicating whether the

19 **System.Web.Services.Description.ServiceDescriptionFormatExtension** is  
20 necessary for the action to which it refers.

21 ServiceDescriptionFormatExtensionCollection class

22 (System.Web.Services.Description)

23 ToString

1  
2  
3 *Description*

4 Represents the collection of extensibility elements used by the Web  
5 Service. This class cannot be inherited.

6 ServiceDescriptionFormatExtensionCollection

7 *Example Syntax:*

8 ToString

9  
10 [C#] public ServiceDescriptionFormatExtensionCollection(object parent);

11 [C++] public: ServiceDescriptionFormatExtensionCollection(Object\* parent);

12 [VB] Public Sub New(ByVal parent As Object)

13 [JScript] public function ServiceDescriptionFormatExtensionCollection(parent :  
14 Object);

15  
16 *Description*

17 Initializes a new instance of the  
18 **System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect**  
19 **ion** class. The **System.Web.Services.Description.ServiceDescription** of which  
20 this collection is a member.

21 Count

22 InnerList

23 Item

24 ToString

*Description*

Gets or sets the value of a member of the  
**System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect  
ion** . The zero-based index of the member in question.

List

Table

Add

[C#] public int Add(object extension);

[C++] public: int Add(Object\* extension);

[VB] Public Function Add(ByVal extension As Object) As Integer

[JScript] public function Add(extension : Object) : int;

*Description*

Adds *extension* to the  
**System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect  
ion** .

*Return Value:* Returns the index value of the member added. The

**System.Web.Services.Description.ServiceDescriptionFormatExtension**, passed  
by reference, to be added to the

**System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect  
ion**.

Contains

1  
2 [C#] public bool Contains(object extension);

3 [C++] public: bool Contains(Object\* extension);

4 [VB] Public Function Contains(ByVal extension As Object) As Boolean

5 [JScript] public function Contains(extension : Object) : Boolean;

6  
7 *Description*

8 Gets a value indicating whether *extension* is a member of the  
9 **System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect**  
10 **ion** .

11 *Return Value:* **true** if the object is a member of the collection; otherwise, **false** .

12 The object that is to be checked whether it is a member of the  
13 **System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect**  
14 **ion**.

15 **CopyTo**

16  
17 [C#] public void CopyTo(object[] array, int index);

18 [C++] public: void CopyTo(Object\* array \_\_gc[], int index);

19 [VB] Public Sub CopyTo(ByVal array() As Object, ByVal index As Integer)

20 [JScript] public function CopyTo(array : Object[], index : int);

21  
22 *Description*

23 Copies the elements in the  
24 **System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect**

ion into *array* , starting at *index* . The destination of the copy action The zero-based index at which to place the first copied element

### Find

[C#] public object Find(Type type);

[C++] public: Object\* Find(Type\* type);

[VB] Public Function Find(ByVal type As Type) As Object

[JScript] public function Find(type : Type) : Object; This method searches the

**System.Web.Services.Description.ServiceDescriptionFormatExtensionCollection** and returns the first member of the collection specified by the parameter passed in.

### Description

This method searches the **System.Web.Services.Description.ServiceDescriptionFormatExtensionCollection** and returns the first instance of *type* that it discovers.

*Return Value:* Returns an object of the specified **System.Type** if its search was successful; **null** otherwise.

Note that this method searches the collection in index order, and returns only the *type* with the lowest-numbered index. A **System.Type** for which to search the collection.

### Find

[C#] public XmlElement Find(string name, string ns);

[C++] public: XmlElement\* Find(String\* name, String\* ns);

1 [VB] Public Function Find(ByVal name As String, ByVal ns As String) As  
2 XmlElement

3 [JScript] public function Find(name : String, ns : String) : XmlElement;

4  
5 *Description*

6 This method searches the  
7 **System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect**  
8 **ion** for a member with name *name* and namespace URI *ns* .

9 *Return Value:* Returns an **System.Xml.XmlElement** if the search is successful;  
10 otherwise **null** .

11 Note that this method searches the collection in index order, and will return  
12 the first **System.Xml.XmlElement** that meets the criteria of the two parameters. A  
13 string representing the name of the **System.Xml.XmlElement** to be found. A  
14 string representing the XML namespace URI of the **System.Xml.XmlElement** to  
15 be found.

16 **FindAll**

17  
18 [C#] public object[] FindAll(Type type);

19 [C++] public: Object\* FindAll(Type\* type) \_\_gc[];

20 [VB] Public Function FindAll(ByVal type As Type) As Object()

21 [JScript] public function FindAll(type : Type) : Object[]; This method searches the  
22 **System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect**  
23 **ion** for all members of the collection specified by the parameter passed in.

24  
25 *Description*



1 This method searches the  
2 **System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect**  
3 **ion** and returns an array of object instances of **System.Type** that it discovers.

4 *Return Value:* Returns an array of **System.Object** instances.

5 The array returned will be empty if the search is unsuccessful. A  
6 **System.Type** for which to search the collection.

#### 7 FindAll

8  
9 [C#] public XmlElement[] FindAll(string name, string ns);

10 [C++] public: XmlElement\* FindAll(String\* name, String\* ns) [];

11 [VB] Public Function FindAll(ByVal name As String, ByVal ns As String) As  
12 XmlElement()

13 [JScript] public function FindAll(name : String, ns : String) : XmlElement[];

#### 15 Description

16 This method searches the  
17 **System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect**  
18 **ion** for members with name *name* and namespace URI *ns* .

19 *Return Value:* Returns an array of **System.Xml.XmlElement** objects.

20 The array returned will be empty if the search is unsuccessful. A string  
21 representing the name attribute of the **System.Xml.XmlElement** objects to be  
22 found. A string representing the XML namespace URI attribute of the  
23 **System.Xml.XmlElement** objects to be found.

#### 24 IndexOf

```

1
2 [C#] public int IndexOf(object extension);
3 [C++] public: int IndexOf(Object* extension);
4 [VB] Public Function IndexOf(ByVal extension As Object) As Integer
5 [JScript] public function IndexOf(extension : Object) : int;
6

```

### *Description*

Gets the zero-based index value of *extension* , the specified member of the **System.Web.Services.Description.ServiceDescriptionFormatExtensionCollection** .

*Return Value:* Returns a 32-bit signed integer. The object for which to return the index value.

### Insert

```

15 [C#] public void Insert(int index, object extension);
16 [C++] public: void Insert(int index, Object* extension);
17 [VB] Public Sub Insert(ByVal index As Integer, ByVal extension As Object)
18 [JScript] public function Insert(index : int, extension : Object);
19

```

### *Description*

Adds *extension* to the **System.Web.Services.Description.ServiceDescriptionFormatExtensionCollection** at the specified *index* . The zero-based index at which to add the new member. The object to add to the

1 **System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect**  
2 **ion.**

3 **IsHandled**

4  
5 [C#] public bool IsHandled(object item);  
6 [C++] public: bool IsHandled(Object\* item);  
7 [VB] Public Function IsHandled(ByVal item As Object) As Boolean  
8 [JScript] public function IsHandled(item : Object) : Boolean;

9  
10 *Description*

11 Gets a value indicating whether *item* is handled.

12 *Return Value:* **true** if *item* is handled; otherwise **false** . An object, either of type  
13 **System.Xml.XmlElement** or  
14 **System.Web.Services.Description.ServiceDescriptionFormatExtension.**

15 **IsRequired**

16  
17 [C#] public bool IsRequired(object item);  
18 [C++] public: bool IsRequired(Object\* item);  
19 [VB] Public Function IsRequired(ByVal item As Object) As Boolean  
20 [JScript] public function IsRequired(item : Object) : Boolean;

21  
22 *Description*

23 Gets a value indicating whether *item* is required.

24 *Return Value:* **true** if *item* is required; otherwise **false** . An object, either of type  
25

1 **System.Xml.XmlElement** or

2 **System.Web.Services.Description.ServiceDescriptionFormatExtension.**

3     OnValidate

4  
5 [C#] protected override void OnValidate(object value);

6 [C++] protected: void OnValidate(Object\* value);

7 [VB] Overrides Protected Sub OnValidate(ByVal value As Object)

8 [JScript] protected override function OnValidate(value : Object);

9  
10 *Description*

11     Performs a check on the **System.Type** of *value* when validating it.

12     This method overrides the base method in order to determine that *value* is  
13 one of the two valid classes, **System.Xml.XmlElement** or  
14 **System.Web.Services.Description.ServiceDescriptionFormatExtension** . The  
15 object to be validated.

16     Remove

17  
18 [C#] public void Remove(object extension);

19 [C++] public: void Remove(Object\* extension);

20 [VB] Public Sub Remove(ByVal extension As Object)

21 [JScript] public function Remove(extension : Object);

22  
23 *Description*

24     Removes the first occurrence of *extension* from the

25 **System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect**

ion . The object to be removed from the

**System.Web.Services.Description.ServiceDescriptionFormatExtensionCollection.**

SetParent

[C#] protected override void SetParent(object value, object parent);

[C++] protected: void SetParent(Object\* value, Object\* parent);

[VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As Object)

[JScript] protected override function SetParent(value : Object, parent : Object);

### *Description*

Sets the parent object of a member of the **System.Web.Services.Description.ServiceDescriptionFormatExtensionCollection** .

This method overrides the base method in order to determine that *value* is of type **System.Web.Services.Description.ServiceDescriptionFormatExtension** . A child object for which to set the parent. The object to be set as parent.

ServiceDescriptionImporter class (System.Web.Services.Description)

ToString

### *Description*

This class exposes methods for generating client proxy classes for Web Services.

ServiceDescriptionImporter

*Example Syntax:*

ToString

[C#] public ServiceDescriptionImporter();

[C++] public: ServiceDescriptionImporter();

[VB] Public Sub New()

[JScript] public function ServiceDescriptionImporter();

#### *Description*

Initializes a new instance of the

**System.Web.Services.Description.ServiceDescriptionImporter** class.

ProtocolName

ToString

[C#] public string ProtocolName {get; set;}

[C++] public: \_\_property String\* get\_ProtocolName();public: \_\_property void  
set\_ProtocolName(String\*);

[VB] Public Property ProtocolName As String

[JScript] public function get ProtocolName() : String;public function set  
ProtocolName(String);

#### *Description*

Gets or sets the name of the protocol to be imported.

Schemas

ToString

```
[C#] public XmlSchemas Schemas {get;}
[C++] public: __property XmlSchemas* get_Schemas();
[VB] Public ReadOnly Property Schemas As XmlSchemas
[JScript] public function get Schemas() : XmlSchemas;
```

#### *Description*

Gets the **System.Xml.Serialization.XmlSchemas** Collection contained in the Web Service to be imported. This property is read-only.

ServiceDescriptions

ToString

```
[C#] public ServiceDescriptionCollection ServiceDescriptions {get;}
[C++] public: __property ServiceDescriptionCollection*
get_ServiceDescriptions();
[VB] Public ReadOnly Property ServiceDescriptions As
ServiceDescriptionCollection
[JScript] public function get ServiceDescriptions() : ServiceDescriptionCollection;
```

#### *Description*

Gets the **System.Web.Services.Description.ServiceDescriptionCollection** being imported by the **ServiceDescriptionImporter**.

As with any collection that forms a read-only property, members can be added to the collection, removed from the collection, or modified using the

1 methods exposed by the collection. However, it is recommended that the  
2 developer use the

3 **System.Web.Services.Description.ServiceDescriptionImporter.AddServiceDe**  
4 **scription(System.Web.Services.Description.ServiceDescription,System.String,**  
5 **System.String)** method to add members to this collection.

6 *Style*

7 *ToString*

8  
9 [C#] public ServiceDescriptionImportStyle Style {get; set;}

10 [C++] public: \_\_property ServiceDescriptionImportStyle get\_Style();public:

11 \_\_property void set\_Style(ServiceDescriptionImportStyle);

12 [VB] Public Property Style As ServiceDescriptionImportStyle

13 [JScript] public function get Style() : ServiceDescriptionImportStyle;public

14 function set Style(ServiceDescriptionImportStyle);

15  
16 *Description*

17 Gets or sets the

18 **System.Web.Services.Description.ServiceDescriptionImportStyle** associated  
19 with the **System.Web.Services.Description.ServiceDescriptionImporter**  
20 instance.

21 *AddServiceDescription*

22  
23 [C#] public void AddServiceDescription(ServiceDescription serviceDescription,  
24 string appSettingUrlKey, string appSettingBaseUrl);

25 [C++] public: void AddServiceDescription(ServiceDescription\*



1 serviceDescription, String\* appSettingUrlKey, String\* appSettingBaseUrl);

2 [VB] Public Sub AddServiceDescription(ByVal serviceDescription As

3 ServiceDescription, ByVal appSettingUrlKey As String, ByVal

4 appSettingBaseUrl As String)

5 [JScript] public function AddServiceDescription(serviceDescription :

6 ServiceDescription, appSettingUrlKey : String, appSettingBaseUrl : String);

7  
8 *Description*

9 Adds the specified **System.Web.Services.Description.ServiceDescription**  
10 to the

11 **System.Web.Services.Description.ServiceDescriptionImporter.ServiceDescrip**  
12 **tions** collection. It also sets the

13 **System.Web.Services.Description.ServiceDescription.AppSettingUrlKey** and

14 **System.Web.Services.Description.ServiceDescription.AppSettingBaseUrl**

15 properties of the **System.Web.Services.Description.ServiceDescription** to be  
16 added.

17 The two string parameters, *appSettingUrlKey* and *appSettingBaseUrl* ,  
18 specify how the **Url** property of the Web Service proxy to be generated from the  
19 imported *serviceDescription* should be constructed. The

20 **System.Web.Services.Description.ServiceDescription** instance to add to the

21 collection Sets the initial value of the **Url** property of the proxy class to be

22 generated from the instance represented by the *serviceDescription* parameter.

23 Specifies that it should be generated from the web.config file's section. Sets the

24 initial value of the **Url** property of the proxy class to be generated from the

25 instance represented by the *serviceDescription* parameter. Specifies that it should

be constructed from a combination of the value of this parameter and the URL specified by the **location** attribute in the WSDL document.

### Import

```
[C#] public ServiceDescriptionImportWarnings Import(CodeNamespace
codeNamespace, CodeCompileUnit codeCompileUnit);
[C++] public: ServiceDescriptionImportWarnings Import(CodeNamespace*
codeNamespace, CodeCompileUnit* codeCompileUnit);
[VB] Public Function Import(ByVal codeNamespace As CodeNamespace, ByVal
codeCompileUnit As CodeCompileUnit) As ServiceDescriptionImportWarnings
[JScript] public function Import(codeNamespace : CodeNamespace,
codeCompileUnit : CodeCompileUnit) : ServiceDescriptionImportWarnings;
```

### Description

Imports a **System.Web.Services.Description.ServiceDescription** from the specified namespace, and generates code for client proxy classes.

*Return Value:* One of the

**System.Web.Services.Description.ServiceDescriptionImportWarnings** values.

This method actually does the work of importing **System.Web.Services.Description.ServiceDescriptions** and **System.Xml.Schema.XmlSchemas**. A **System.CodeDom.CodeNamespace** instance that determines the namespace of the **System.Web.Services.Description.ServiceDescription** to be imported. A **System.CodeDom.CodeCompileUnit** instance that will add references to the appropriate assemblies.

ServiceDescriptionImportStyle enumeration

(System.Web.Services.Description)

ToString

*Description*

Specifies whether the Import is made to the server or the client machine.

ToString

[C#] public const ServiceDescriptionImportStyle Client;

[C++] public: const ServiceDescriptionImportStyle Client;

[VB] Public Const Client As ServiceDescriptionImportStyle

[JScript] public var Client : ServiceDescriptionImportStyle;

*Description*

Specifies that the Import should be made to the client machine.

ToString

[C#] public const ServiceDescriptionImportStyle Server;

[C++] public: const ServiceDescriptionImportStyle Server;

[VB] Public Const Server As ServiceDescriptionImportStyle

[JScript] public var Server : ServiceDescriptionImportStyle;

*Description*

Specifies that the Import should be made to the server.

ServiceDescriptionImportWarnings enumeration

(System.Web.Services.Description)

ToString

*Description*

Specifies the type of warnings produced by

**System.Web.Services.Description.ServiceDescriptionImporter.Import(System  
.CodeDom.CodeNamespace, System.CodeDom.CodeCompileUnit) .**

ToString

[C#] public const ServiceDescriptionImportWarnings NoCodeGenerated;

[C++] public: const ServiceDescriptionImportWarnings NoCodeGenerated;

[VB] Public Const NoCodeGenerated As ServiceDescriptionImportWarnings

[JScript] public var NoCodeGenerated : ServiceDescriptionImportWarnings;

*Description*

Specifies that no proxy class was generated by the

**System.Web.Services.Description.ServiceDescriptionImporter.Import(System  
.CodeDom.CodeNamespace, System.CodeDom.CodeCompileUnit) method.**

ToString

[C#] public const ServiceDescriptionImportWarnings NoMethodsGenerated;

[C++] public: const ServiceDescriptionImportWarnings NoMethodsGenerated;

[VB] Public Const NoMethodsGenerated As ServiceDescriptionImportWarnings

1 [JScript] public var NoMethodsGenerated : ServiceDescriptionImportWarnings;

3 *Description*

4 Specifies that the proxy class generated by the  
5 **System.Web.Services.Description.ServiceDescriptionImporter.Import(System**  
6 **.CodeDom.CodeNamespace,System.CodeDom.CodeCompileUnit)** method  
7 includes no methods.

8 ToString

10 [C#] public const ServiceDescriptionImportWarnings OptionalExtensionsIgnored;

11 [C++] public: const ServiceDescriptionImportWarnings

12 OptionalExtensionsIgnored;

13 [VB] Public Const OptionalExtensionsIgnored As

14 ServiceDescriptionImportWarnings

15 [JScript] public var OptionalExtensionsIgnored :

16 ServiceDescriptionImportWarnings;

18 *Description*

19 Specifies that at least one optional  
20 **System.Web.Services.Description.ServiceDescriptionFormatExtension** for the  
21 **System.Web.Services.Description.ServiceDescription** to be imported has been  
22 ignored.

23 ToString

25 [C#] public const ServiceDescriptionImportWarnings RequiredExtensionsIgnored;

1 [C++] public: const ServiceDescriptionImportWarnings

2 RequiredExtensionsIgnored;

3 [VB] Public Const RequiredExtensionsIgnored As

4 ServiceDescriptionImportWarnings

5 [JScript] public var RequiredExtensionsIgnored :

6 ServiceDescriptionImportWarnings;

7  
8 *Description*

9 Specifies that at least one necessary

10 **System.Web.Services.Description.ServiceDescriptionFormatExtension** for the

11 **System.Web.Services.Description.ServiceDescription** to be imported has been

12 ignored.

13 ToString

14  
15 [C#] public const ServiceDescriptionImportWarnings

16 UnsupportedBindingsIgnored;

17 [C++] public: const ServiceDescriptionImportWarnings

18 UnsupportedBindingsIgnored;

19 [VB] Public Const UnsupportedBindingsIgnored As

20 ServiceDescriptionImportWarnings

21 [JScript] public var UnsupportedBindingsIgnored :

22 ServiceDescriptionImportWarnings;

23  
24 *Description*

1 Specifies that at least one **System.Web.Services.Description.Binding** for  
2 the **System.Web.Services.Description.ServiceDescription** to be imported is of  
3 an unsupported type and has been ignored.

4 ToString

5  
6 [C#] public const ServiceDescriptionImportWarnings

7 UnsupportedOperationsIgnored;

8 [C++] public: const ServiceDescriptionImportWarnings

9 UnsupportedOperationsIgnored;

10 [VB] Public Const UnsupportedOperationsIgnored As

11 ServiceDescriptionImportWarnings

12 [JScript] public var UnsupportedOperationsIgnored :

13 ServiceDescriptionImportWarnings;

14  
15 *Description*

16 Specifies that at least one **System.Web.Services.Description.Operation**  
17 for the **System.Web.Services.Description.ServiceDescription** to be imported is  
18 of an unsupported type and has been ignored.

19 ServiceDescriptionReflector class (System.Web.Services.Description)

20 ToString

21  
22  
23 *Description*

24  
25 ServiceDescriptionReflector

*Example Syntax:*

ToString

[C#] public ServiceDescriptionReflector();

[C++] public: ServiceDescriptionReflector();

[VB] Public Sub New()

[JScript] public function ServiceDescriptionReflector();

*Description*

Schemas

ToString

[C#] public XmlSchemas Schemas {get;}

[C++] public: \_\_property XmlSchemas\* get\_Schemas();

[VB] Public ReadOnly Property Schemas As XmlSchemas

[JScript] public function get Schemas() : XmlSchemas;

*Description*

ServiceDescriptions

ToString

[C#] public ServiceDescriptionCollection ServiceDescriptions {get;}

[C++] public: \_\_property ServiceDescriptionCollection\*



```

1  get_ServiceDescriptions();
2  [VB] Public ReadOnly Property ServiceDescriptions As
3  ServiceDescriptionCollection
4  [JScript] public function get ServiceDescriptions() : ServiceDescriptionCollection;

```

6 | *Description*

8 Reflect

```
10 [C#] public void Reflect(Type type, string url);
```

```
11  [C++] public: void Reflect(Type* type, String* url);
```

```
12 [VB] Public Sub Reflect(ByVal type As Type, ByVal url As String)
```

```
13 [JScript] public function Reflect(type : Type, url : String);
```

15 *Description*

17 SoapAddressBinding class (System.Web.Services.Description)

18 ToString

21 | *Description*

22 Represents an extensibility element added to a

```
23 System.Web.Services.Description.Port within a Web Service. This class cannot
24 be inherited.
```

25 SoapAddressBinding

*Example Syntax:*

ToString

[C#] public SoapAddressBinding();

[C++] public: SoapAddressBinding();

[VB] Public Sub New()

[JScript] public function SoapAddressBinding();

Handled

Location

ToString

*Description*

Gets or sets a value representing the URI for the **System.Web.Services.Description.Port** to which the **SoapAddressBinding** applies.

Parent

Required

SoapBinding class (System.Web.Services.Description)

ToString

*Description*

Represents an extension added to a **System.Web.Services.Description.Binding** within a Web Service. It specifies

that the data transmission will use SOAP (Simple Object Access Protocol). This class cannot be inherited.

ToString

[C#] public const string HttpTransport;

[C++] public: const String\* HttpTransport;

[VB] Public Const HttpTransport As String

[JScript] public var HttpTransport : String;

*Description*

ToString

[C#] public const string Namespace;

[C++] public: const String\* Namespace;

[VB] Public Const Namespace As String

[JScript] public var Namespace : String;

*Description*

Gets the URI (<http://schemas.xmlsoap.org/wsdl/soap/>) for the XML namespace of the **SoapBinding** class. This field is constant.

SoapBinding

*Example Syntax:*

ToString

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#] public SoapBinding();  
[C++] public: SoapBinding();  
[VB] Public Sub New()  
[JScript] public function SoapBinding();
```

- Handled
- Parent
- Required
- Style
- ToString

*Description*

Specifies the type of SOAP binding used by the **SoapBinding** instance.

- Transport
- ToString

```
[C#] public string Transport {get; set;}  
[C++] public: __property String* get_Transport();public: __property void  
set_Transport(String*);  
[VB] Public Property Transport As String  
[JScript] public function get Transport() : String;public function set  
Transport(String);
```

*Description*

1 Gets or sets the URI for the specification for HTTP transmission of SOAP  
2 data.

3 In general usage, this required property value will be the same as that of the  
4 **System.Web.Services.Description.SoapBinding.HttpTransport** field. However,  
5 this property value may be any URI for data-transmission specifications, such as  
6 SMTP, FTP, etc.

7 SoapBindingStyle enumeration (System.Web.Services.Description)

8 ToString

9  
10  
11 *Description*

12 Specifies the type of action that occurs in the Web Service at the level to  
13 which this enumeration is applied.

14 This enumeration allows the user to specify whether the message being  
15 transmitted is procedure- or document-oriented. Procedure-oriented messages  
16 contain parameters and return values, while document-oriented messages contain  
17 documents.

18 ToString

19  
20 [C#] public const SoapBindingStyle Default;

21 [C++] public: const SoapBindingStyle Default;

22 [VB] Public Const Default As SoapBindingStyle

23 [JScript] public var Default : SoapBindingStyle;

24  
25 *Description*

1 This value specifies that the action should use the default value appropriate  
2 to that level of the WSDL file.

3 ToString

4  
5 [C#] public const SoapBindingStyle Document;

6 [C++] public: const SoapBindingStyle Document;

7 [VB] Public Const Document As SoapBindingStyle

8 [JScript] public var Document : SoapBindingStyle;

9  
10 *Description*

11 Specifies that the message to be transmitted contains documents.

12 ToString

13  
14 [C#] public const SoapBindingStyle Rpc;

15 [C++] public: const SoapBindingStyle Rpc;

16 [VB] Public Const Rpc As SoapBindingStyle

17 [JScript] public var Rpc : SoapBindingStyle;

18  
19 *Description*

20 Specifies that the message to be transmitted contains a call to a procedure.

21 RPC is an acronym for "remote procedure call."

22 SoapBindingUse enumeration (System.Web.Services.Description)

23 ToString

TOP SECRET

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

*Description*

Specifies whether the message parts are encoded using given rules, or  
define a concrete schema for the message.

ToString

[C#] public const SoapBindingUse Default;  
[C++] public: const SoapBindingUse Default;  
[VB] Public Const Default As SoapBindingUse  
[JScript] public var Default : SoapBindingUse;

*Description*

Specifies an empty string ("" ) value for the corresponding XML attribute.

ToString

[C#] public const SoapBindingUse Encoded;  
[C++] public: const SoapBindingUse Encoded;  
[VB] Public Const Encoded As SoapBindingUse  
[JScript] public var Encoded : SoapBindingUse;

*Description*

Specifies that the message parts are encoded using given encoding rules.

ToString

```

1
2 [C#] public const SoapBindingUse Literal;
3 [C++] public: const SoapBindingUse Literal;
4 [VB] Public Const Literal As SoapBindingUse
5 [JScript] public var Literal : SoapBindingUse;
6

```

#### *Description*

Specifies that the message parts represent a concrete schema.

SoapBodyBinding class (System.Web.Services.Description)

ToString

#### *Description*

Represents an extensibility element added to an **System.Web.Services.Description.InputBinding** or an **System.Web.Services.Description.OutputBinding**. It specifies how messages, either abstract type definitions or concrete schema definitions, appear within the SOAP body element of the transmission. This class cannot be inherited.

SoapBodyBinding

#### *Example Syntax:*

ToString

```

23 [C#] public SoapBodyBinding();
24 [C++] public: SoapBodyBinding();
25

```



1 [VB] Public Sub New()

2 [JScript] public function SoapBodyBinding();

3     Encoding

4     ToString

6 [C#] public string Encoding {get; set;}

7 [C++] public: \_\_property String\* get\_Encoding();public: \_\_property void  
8 set\_Encoding(String\*);

9 [VB] Public Property Encoding As String

10 [JScript] public function get Encoding() : String;public function set  
11 Encoding(String);

13 *Description*

14     Supplies a list of space-delimited URIs representing the encoding style (or  
15 styles) to be used to encode the messages within the SOAP body.

16     The value of this property should be set only if the value of the  
17 **System.Web.Services.Description.S SoapBodyBinding.Use** property is **Encoded** .

18     Handled

19     Namespace

20     ToString

23 *Description*

24     Get or sets the URI for encoding of content not specifically defined by the  
25 **System.Web.Services.Description.S SoapBodyBinding.Encoding** property.

This property will return an empty string ("") if the property value has not been set. The value should only be set if the value of the **System.Web.Services.Description.SoapBodyBinding.Use** property is **Encoded**.

Parent

Parts

ToString

#### *Description*

Indicates which parts of the transmitted message appear somewhere within the SOAP body portion of that transmission.

Some parts of the transmitted message can appear in portions other than the SOAP body, such as when SOAP is used with a

**System.Web.Services.Description.MimeMultipartRelatedBinding**.

PartsString

ToString

[C#] public string PartsString {get; set;}

[C++] public: \_\_property String\* get\_PartsString();public: \_\_property void set\_PartsString(String\*);

[VB] Public Property PartsString As String

[JScript] public function get PartsString() : String;public function set PartsString(String);

#### *Description*

1 Indicates which parts of the transmitted message appear somewhere within  
2 the SOAP body portion of that transmission.

3 Some parts of the transmitted message can appear in portions other than the  
4 SOAP body, such as when SOAP is used with a  
5 **System.Web.Services.Description.MimeMultipartRelatedBinding** .

6 Required

7 Use

8 ToString

9  
10  
11 *Description*

12 Indicates whether the message parts are encoded based on specified  
13 encoding rules, or define the concrete schema of the message.

14 The **System.Web.Services.Description.SoapBodyBinding.Namespace**  
15 and **System.Web.Services.Description.SoapBodyBinding.Encoding** properties  
16 depend on the value of this property. Their values should be set only if the value  
17 of this property is **Encoded** .

18 SoapExtensionImporter class (System.Web.Services.Description)

19 ToString

20  
21  
22 *Description*

23  
24 SoapExtensionImporter

25 *Example Syntax:*

1 ToString

2  
3 [C#] protected SoapExtensionImporter();

4 [C++] protected: SoapExtensionImporter();

5 [VB] Protected Sub New()

6 [JScript] protected function SoapExtensionImporter();

7 ImportContext

8 ToString

9  
10 [C#] public SoapProtocolImporter ImportContext {get; set;}

11 [C++] public: \_\_property SoapProtocolImporter\* get\_ImportContext();public:

12 \_\_property void set\_ImportContext(SoapProtocolImporter\*);

13 [VB] Public Property ImportContext As SoapProtocolImporter

14 [JScript] public function get ImportContext() : SoapProtocolImporter;public

15 function set ImportContext(SoapProtocolImporter);

16  
17 *Description*

18  
19 ImportMethod

20  
21 [C#] public abstract void ImportMethod(CodeAttributeDeclarationCollection

22 metadata);

23 [C++] public: virtual void ImportMethod(CodeAttributeDeclarationCollection\*

24 metadata) = 0;

25 [VB] MustOverride Public Sub ImportMethod(ByVal metadata As

CodeAttributeDeclarationCollection)

[JScript] public abstract function ImportMethod(metadata :

CodeAttributeDeclarationCollection);

*Description*

SoapExtensionReflector class (System.Web.Services.Description)

ToString

*Description*

SoapExtensionReflector

*Example Syntax:*

ToString

[C#] protected SoapExtensionReflector();

[C++] protected: SoapExtensionReflector();

[VB] Protected Sub New()

[JScript] protected function SoapExtensionReflector();

ReflectionContext

ToString

[C#] public ProtocolReflector ReflectionContext {get; set;}

[C++] public: \_\_property ProtocolReflector\* get\_ReflectionContext();public:

1 \_\_property void set\_ReflectionContext(ProtocolReflector\*);  
 2 [VB] Public Property ReflectionContext As ProtocolReflector  
 3 [JScript] public function get ReflectionContext() : ProtocolReflector;public  
 4 function set ReflectionContext(ProtocolReflector);

5  
 6 *Description*

7  
 8 ReflectMethod

9  
 10 [C#] public abstract void ReflectMethod();  
 11 [C++] public: virtual void ReflectMethod() = 0;  
 12 [VB] MustOverride Public Sub ReflectMethod()  
 13 [JScript] public abstract function ReflectMethod();

14  
 15 *Description*

16  
 17 SoapFaultBinding class (System.Web.Services.Description)  
 18 ToString

19  
 20  
 21 *Description*

22 Represents an extensibility element added to a  
 23 **System.Web.Services.Description.FaultBinding** within a Web Service. It  
 24 specifies the contents of any SOAP fault message returned. This class cannot be  
 25 inherited.

SoapFaultBinding

*Example Syntax:*

ToString

[C#] public SoapFaultBinding();

[C++] public: SoapFaultBinding();

[VB] Public Sub New()

[JScript] public function SoapFaultBinding();

Encoding

ToString

[C#] public string Encoding {get; set;}

[C++] public: \_\_property String\* get\_Encoding();public: \_\_property void

set\_Encoding(String\*);

[VB] Public Property Encoding As String

[JScript] public function get Encoding() : String;public function set

Encoding(String);

*Description*

Gets or sets a URI representing the encoding style used to encode the SOAP fault message.

The value of this property should be set only if the value of the **System.Web.Services.Description.SoapFaultBinding.Use** property is **Encoded**.

Handled

Namespace

ToString

*Description*

Get or sets the URI for encoding of content not specifically defined by the **System.Web.Services.Description.SoapFaultBinding.Encoding** property.

This property will return an empty string ("") if the property value has not been set. The value should only be set if the value of the **System.Web.Services.Description.SoapFaultBinding.Use** property is **Encoded**.

Parent

Required

Use

ToString

*Description*

Specifies whether the fault message is encoded using encoding rules specified by the **System.Web.Services.Description.SoapFaultBinding.Encoding** property, or is encapsulated within a concrete schema.

SoapHeaderBinding class (System.Web.Services.Description)

ToString

*Description*



Represents an extensibility element added to an **System.Web.Services.Description.InputBinding** or an **System.Web.Services.Description.OutputBinding** within a Web Service. This class cannot be inherited.

SoapHeaderBinding

*Example Syntax:*

ToString

[C#] public SoapHeaderBinding();

[C++] public: SoapHeaderBinding();

[VB] Public Sub New()

[JScript] public function SoapHeaderBinding();

Encoding

ToString

[C#] public string Encoding {get; set;}

[C++] public: \_\_property String\* get\_Encoding();public: \_\_property void set\_Encoding(String\*);

[VB] Public Property Encoding As String

[JScript] public function get Encoding() : String;public function set Encoding(String);

### *Description*

Gets or sets a URI representing the encoding style used to encode the SOAP header.

The value of this property should be set only if the value of the **System.Web.Services.Description.SoapHeaderBinding.Use** property is

**Encoded .**

Handled

MapToProperty

ToString

*Description*

Message

ToString

[C#] public XmlQualifiedName Message {get; set;}

[C++] public: \_\_property XmlQualifiedName\* get\_Message();public: \_\_property

void set\_Message(XmlQualifiedName\*);

[VB] Public Property Message As XmlQualifiedName

[JScript] public function get Message() : XmlQualifiedName;public function set

Message(XmlQualifiedName);

*Description*

Gets or sets a value specifying the name of the

**System.Web.Services.Description.Message** within the Web Service to which the **SoapHeaderBinding** applies.

The default value of this property is an empty string ("").

1        Namespace

2        ToString

3

4    [C#] public string Namespace {get; set;}

5    [C++] public: \_\_property String\* get\_Namespace();public: \_\_property void

6    set\_Namespace(String\*);

7    [VB] Public Property Namespace As String

8    [JScript] public function get Namespace() : String;public function set

9    Namespace(String);

10

11    *Description*

12        Get or sets the URI for encoding of content not specifically defined by the

13    **System.Web.Services.Description.SoapHeaderBinding.Encoding** property.

14        This property will return an empty string ("") if the property value has not

15    been set. The value should only be set if the value of the

16    **System.Web.Services.Description.SoapHeaderBinding.Use** property is

17    **Encoded .**

18        Parent

19        Part

20        ToString

21

22

23    *Description*

24

25

1 Gets or sets a value indicating which  
2 **System.Web.Services.Description.MessagePart** within the Web Service the  
3 **SoapHeaderBinding** applies to.

4 Required

5 Use

6 ToString

7  
8  
9 *Description*

10 Specifies whether the header is encoded using encoding rules specified by  
11 the **System.Web.Services.Description.SoapHeaderBinding.Encoding** property,  
12 or is encapsulated within a concrete schema.

13 SoapHeaderFaultBinding class (System.Web.Services.Description)

14 ToString

15  
16  
17 *Description*

18 Represents an extensibility element added to an  
19 **System.Web.Services.Description.InputBinding** or an  
20 **System.Web.Services.Description.OutputBinding** within the WebService. It  
21 specifies the SOAP header types used to transmit error information within the  
22 SOAP header. This class cannot be inherited.

23 SoapHeaderFaultBinding

24 *Example Syntax:*

25 ToString

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#] public SoapHeaderFaultBinding();
[C++] public: SoapHeaderFaultBinding();
[VB] Public Sub New()
[JavaScript] public function SoapHeaderFaultBinding();
    Encoding
    ToString

[C#] public string Encoding {get; set;}
[C++] public: __property String* get_Encoding();public: __property void
set_Encoding(String*);
[VB] Public Property Encoding As String
[JavaScript] public function get Encoding() : String;public function set
Encoding(String);
```

*Description*

Gets or sets a URI representing the encoding style used to encode the error message for the SOAP header.

The value of this property should be set only if the value of the **System.Web.Services.Description.S SoapHeaderFaultBinding.Use** property is **Encoded**.

- Handled
- Message
- ToString

1  
2  
3 *Description*

4 Gets or sets a value specifying the name of the  
5 **System.Web.Services.Description.Message** within the Web Service to which the  
6 **SoapHeaderFaultBinding** applies.

7 The default value of this property is an empty string ("").

8 Namespace

9 ToString

10  
11 [C#] public string Namespace {get; set;}

12 [C++] public: \_\_property String\* get\_Namespace();public: \_\_property void  
13 set\_Namespace(String\*);

14 [VB] Public Property Namespace As String

15 [JScript] public function get Namespace() : String;public function set  
16 Namespace(String);

17  
18 *Description*

19 Get or sets the URI for encoding of content not specifically defined by the  
20 **System.Web.Services.Description.SoapHeaderFaultBinding.Encoding**  
21 property.

22 This property will return an empty string ("") if the property value has not  
23 been set. The value should only be set if the value of the  
24 **System.Web.Services.Description.SoapHeaderBinding.Use** property is  
25 **Encoded** .

Parent

Part

ToString

*Description*

Gets or sets a value indicating which **System.Web.Services.Description.MessagePart** within the Web Service the **SoapHeaderFaultBinding** applies to.

Required

Use

ToString

*Description*

Specifies whether the header is encoded using encoding rules specified by the **System.Web.Services.Description.SoapHeaderBinding.Encoding** property, or is encapsulated within a concrete schema.

SoapOperationBinding class (System.Web.Services.Description)

ToString

*Description*

Represents an extension added to an **System.Web.Services.Description.OperationBinding** within a Web Service.

1 Specifies that the message transmission will be in SOAP format. This class cannot  
2 be inherited.

3 SoapOperationBinding

4 *Example Syntax:*

5 ToString

6  
7 [C#] public SoapOperationBinding();

8 [C++] public: SoapOperationBinding();

9 [VB] Public Sub New()

10 [JScript] public function SoapOperationBinding();

11 Handled

12 Parent

13 Required

14 SoapAction

15 ToString

16  
17  
18 *Description*

19 Specifies the URI for the SOAP header.

20 This property is required for HTTP protocol binding of SOAP.

21 Style

22 ToString

23  
24 [C#] public SoapBindingStyle Style {get; set;}

25 [C++] public: \_\_property SoapBindingStyle get\_Style();public: \_\_property void



```

1 set_Style(SoapBindingStyle);
2 [VB] Public Property Style As SoapBindingStyle
3 [JScript] public function get Style() : SoapBindingStyle;public function set
4 Style(SoapBindingStyle);
5

```

#### *Description*

One of the **System.Web.Services.Description.SoapBindingStyle** values.

The default is **Document** .

SoapProtocolImporter class (System.Web.Services.Description)

ToString

#### *Description*

SoapProtocolImporter

*Example Syntax:*

ToString

```

19 [C#] public SoapProtocolImporter();
20 [C++] public: SoapProtocolImporter();
21 [VB] Public Sub New()
22 [JScript] public function SoapProtocolImporter();
23
24 AbstractSchemas
25
26 Binding
27
28 ClassName

```

1	ClassNames
2	CodeNamespace
3	CodeTypeDeclaration
4	ConcreteSchemas
5	InputMessage
6	MethodName
7	Operation
8	OperationBinding
9	OutputMessage
10	Port
11	PortType
12	ProtocolName
13	ToString

16 *Description*

18	Schemas
19	Service
20	ServiceDescriptions
21	SoapBinding
22	ToString

25 *Description*

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

SoapExporter

ToString

[C#] public SoapCodeExporter SoapExporter {get;}

[C++] public: \_\_property SoapCodeExporter\* get\_SoapExporter();

[VB] Public ReadOnly Property SoapExporter As SoapCodeExporter

[JScript] public function get SoapExporter() : SoapCodeExporter;

*Description*

SoapImporter

ToString

[C#] public SoapSchemaImporter SoapImporter {get;}

[C++] public: \_\_property SoapSchemaImporter\* get\_SoapImporter();

[VB] Public ReadOnly Property SoapImporter As SoapSchemaImporter

[JScript] public function get SoapImporter() : SoapSchemaImporter;

*Description*

Style

Warnings

XmlExporter

ToString

1  
2  
3 *Description*

4  
5 XmlImporter

6 ToString

7  
8 [C#] public XmlSchemaImporter XmlImporter {get;}

9 [C++] public: \_\_property XmlSchemaImporter\* get\_XmlImporter();

10 [VB] Public ReadOnly Property XmlImporter As XmlSchemaImporter

11 [JScript] public function get XmlImporter() : XmlSchemaImporter;

12  
13 *Description*

14  
15 BeginClass

16  
17 [C#] protected override CodeTypeDeclaration BeginClass();

18 [C++] protected: CodeTypeDeclaration\* BeginClass();

19 [VB] Overrides Protected Function BeginClass() As CodeTypeDeclaration

20 [JScript] protected override function BeginClass() : CodeTypeDeclaration;

21  
22 *Description*

23  
24 BeginNamespace

1  
2 [C#] protected override void BeginNamespace();  
3 [C++] protected: void BeginNamespace();  
4 [VB] Overrides Protected Sub BeginNamespace()  
5 [JScript] protected override function BeginNamespace();

6  
7 *Description*

8  
9 EndClass

10  
11 [C#] protected override void EndClass();  
12 [C++] protected: void EndClass();  
13 [VB] Overrides Protected Sub EndClass()  
14 [JScript] protected override function EndClass();

15  
16 *Description*

17  
18 EndNamespace

19  
20 [C#] protected override void EndNamespace();  
21 [C++] protected: void EndNamespace();  
22 [VB] Overrides Protected Sub EndNamespace()  
23 [JScript] protected override function EndNamespace();

24  
25 *Description*

## GenerateMethod

[C#] protected override CodeMemberMethod GenerateMethod();  
[C++] protected: CodeMemberMethod\* GenerateMethod();  
[VB] Overrides Protected Function GenerateMethod() As CodeMemberMethod  
[JScript] protected override function GenerateMethod() : CodeMemberMethod;

### *Description*

## IsBindingSupported

[C#] protected override bool IsBindingSupported();  
[C++] protected: bool IsBindingSupported();  
[VB] Overrides Protected Function IsBindingSupported() As Boolean  
[JScript] protected override function IsBindingSupported() : Boolean;

### *Description*

## IsOperationFlowSupported

[C#] protected override bool IsOperationFlowSupported(OperationFlow flow);  
[C++] protected: bool IsOperationFlowSupported(OperationFlow flow);  
[VB] Overrides Protected Function IsOperationFlowSupported(ByVal flow As  
OperationFlow) As Boolean

1 [JScript] protected override function IsOperationFlowSupported(flow :  
2 OperationFlow) : Boolean;

3  
4 *Description*

5  
6 SoapTransportImporter class (System.Web.Services.Description)

7 UnsupportedOperationWarning

8  
9  
10 *Description*

11  
12 SoapTransportImporter

13 *Example Syntax:*

14 UnsupportedOperationWarning

15  
16 [C#] protected SoapTransportImporter();

17 [C++] protected: SoapTransportImporter();

18 [VB] Protected Sub New()

19 [JScript] protected function SoapTransportImporter();

20 ImportContext

21 UnsupportedOperationWarning

22  
23 [C#] public SoapProtocolImporter ImportContext {get; set;}

24 [C++] public: \_\_property SoapProtocolImporter\* get\_ImportContext();public:

25 \_\_property void set\_ImportContext(SoapProtocolImporter\*);

1 [VB] Public Property ImportContext As SoapProtocolImporter  
2 [JScript] public function get ImportContext() : SoapProtocolImporter;public  
3 function set ImportContext(SoapProtocolImporter);  
4

5 *Description*  
6

7 ImportClass  
8

9 [C#] public abstract void ImportClass();  
10 [C++] public: virtual void ImportClass() = 0;  
11 [VB] MustOverride Public Sub ImportClass()  
12 [JScript] public abstract function ImportClass();  
13

14 *Description*  
15

16 IsSupportedTransport  
17

18 [C#] public abstract bool IsSupportedTransport(string transport);  
19 [C++] public: virtual bool IsSupportedTransport(String\* transport) = 0;  
20 [VB] MustOverride Public Function IsSupportedTransport(ByVal transport As  
21 String) As Boolean  
22 [JScript] public abstract function IsSupportedTransport(transport : String) :  
23 Boolean;  
24

25 *Description*



Types class (System.Web.Services.Description)

ToString

### *Description*

Derived from the **System.Web.Services.Description.DocumentableItem** class, this class describes data type definitions relevant to exchanged messages.

This class cannot be inherited.

This class is a container for datatype definitions for the Web Service. Note especially that this class does NOT represent a collection of **System.Type** objects, as its name might imply.

Types

*Example Syntax:*

ToString

[C#] public Types();

[C++] public: Types();

[VB] Public Sub New()

[JScript] public function Types();

Documentation

Extensions

ToString

1  
2  
3 *Description*

4 Gets the collection of ServiceDescriptionFormatExtension elements  
5 included in the Web Service. This property is read-only.

6 In the default implementation of the  
7  
8  
9

10 **System.Web.Services.Discovery**

11  
12 *Description*

13 The **System.Web.Services.Discovery** namespace consists of the classes  
14 that allows Web Service consumers to locate the available Web Services on a Web  
15 server through a process called Web Services Discovery.

16 ContractReference class (System.Web.Services.Discovery)  
17  
18

19 *Description*

20 Represents a reference in a discovery document to a Service Description.

21 Web Services discovery involves discovering the available Web Services  
22 given an URL. The URL usually points to a discovery document, which typically  
23 has a .disco file extension. Within a discovery document are references to  
24 information about the existence of Web Services. These references can refer to  
25

Service Descriptions, XML Schema Definition language (XSD) schemas or other discovery documents. This class represents a reference to a Service Description.

[C#] public const string Namespace;

[C++] public: const String\* Namespace;

[VB] Public Const Namespace As String

[JScript] public var Namespace : String;

### *Description*

XML namespace for Service Description references in discovery documents.

Within a discovery document, a reference to a Service Description is contained within a **contractRef** XML element, which is a part of the XML namespace specified in the

**System.Web.Services.Discovery.ContractReference.Namespace** constant.

Constructors:

ContractReference

*Example Syntax:*

[C#] public ContractReference();

[C++] public: ContractReference();

[VB] Public Sub New()

[JScript] public function ContractReference(); Initializes a new instance of the

**System.Web.Services.Discovery.ContractReference** class.

## Description

Initializes a new instance of the

**System.Web.Services.Discovery.ContractReference** class using default values.

Use this constructor to create and initialize a new instance of the

**System.Web.Services.Discovery.ContractReference** class using default values.

ContractReference

### Example Syntax:

[C#] public ContractReference(string href);

[C++] public: ContractReference(String\* href);

[VB] Public Sub New(ByVal href As String)

[JScript] public function ContractReference(href : String);

## Description

Initializes a new instance of the

**System.Web.Services.Discovery.ContractReference** class using the supplied reference to a Service Description.

Use this constructor to create and initialize a new instance of the

**System.Web.Services.Discovery.ContractReference** class using the specified reference name. The URL for a Service Description. Initializes the

**System.Web.Services.Discovery.ContractReference.Ref** property value.

ContractReference

### Example Syntax:

```

1
2 [C#] public ContractReference(string href, string docRef);
3 [C++] public: ContractReference(String* href, String* docRef);
4 [VB] Public Sub New(ByVal href As String, ByVal docRef As String)
5 [JScript] public function ContractReference(href : String, docRef : String);
6

```

### *Description*

Initializes a new instance of the **System.Web.Services.Discovery.ContractReference** class using the supplied reference to a Service Description and a Web Service implementing the Service Description.

Use this constructor to create and initialize a new instance of the **System.Web.Services.Discovery.ContractReference** class using the specified reference name and docRef. The URL for a Service Description. Initializes the **System.Web.Services.Discovery.ContractReference.Ref** property value. The URL for a Web Service implementing the Service Description at *href*. Initializes the **System.Web.Services.Discovery.ContractReference.DocRef** property value.

### Properties:

ClientProtocol

Contract

### *Description*

Gets a **System.Web.Services.Description.ServiceDescription** object representing the Service Description.

## DefaultFilename

[C#] public override string DefaultFilename {get;}

[C++] public: \_\_property virtual String\* get\_DefaultFilename();

[VB] Overrides Public ReadOnly Property DefaultFilename As String

[JScript] public function get DefaultFilename() : String;

### *Description*

Gets the name of the file to use by default when saving the referenced Service Description.

## DocRef

[C#] public string DocRef {get; set;}

[C++] public: \_\_property String\* get\_DocRef();public: \_\_property void set\_DocRef(String\*);

[VB] Public Property DocRef As String

[JScript] public function get DocRef() : String;public function set DocRef(String);

### *Description*

Gets and sets the URL for a Web Service implementing the Service Description referenced in the

**System.Web.Services.Discovery.ContractReference.Ref** property.

Within a discovery document, a reference to a Service Description is contained within a **contractRef** XML element. The **contractRef** XML element has two attributes described in the following table.

## Ref

[C#] public string Ref {get; set;}

[C++] public: \_\_property String\* get\_Ref();public: \_\_property void  
set\_Ref(String\*);

[VB] Public Property Ref As String

[JScript] public function get Ref() : String;public function set Ref(String);

### *Description*

Gets or sets the URL to the referenced Service Description.

Within a discovery document, a reference to a Service Description is contained within a contractRef XML element. The contractRef XML element has two attributes described in the following table.

## Url

[C#] public override string Url {get; set;}

[C++] public: \_\_property virtual String\* get\_Url();public: \_\_property virtual void  
set\_Url(String\*);

[VB] Overrides Public Property Url As String

[JScript] public function get Url() : String;public function set Url(String);

### *Description*

Gets or sets the URL for the referenced Service Description.

For the **System.Web.Services.Discovery.ContractReference** class, the **System.Web.Services.Discovery.ContractReference.Url** property returns the value of the **System.Web.Services.Discovery.ContractReference.Ref** property.

Methods:

**ReadDocument**

[C#] public override object ReadDocument(Stream stream);

[C++] public: Object\* ReadDocument(Stream\* stream);

[VB] Overrides Public Function ReadDocument(ByVal stream As Stream) As Object

[JScript] public override function ReadDocument(stream : Stream) : Object;

### *Description*

Reads the Service Description from the passed **System.IO.Stream** and returns the Service Description.

*Return Value:* A **System.Web.Services.Description.ServiceDescription** containing the contents of the referenced Service Description. **System.IO.Stream** containing the Service Description.

**Resolve**

[C#] protected internal override void Resolve(string contentType, Stream stream);

[C++] protected public: void Resolve(String\* contentType, Stream\* stream);

[VB] Overrides Protected Friend Dim Sub Resolve(ByVal contentType As String, ByVal stream As Stream)

[JScript] package override function Resolve(contentType : String, stream :



Stream);

### Description

Resolves whether the the referenced document is valid.

If the MIME type is **text/xml** and the contents of *stream* is a Service Description, then the contents of *stream* are added to the **System.Web.Services.Discovery.DiscoveryClientProtocol.References** and **System.Web.Services.Discovery.DiscoveryClientProtocol.Documents** properties of **System.Web.Services.Discovery.DiscoveryReference.ClientProtocol** . The MIME content type of *stream*. The **System.IO.Stream** containing the referenced document.

### WriteDocument

[C#] public override void WriteDocument(object document, Stream stream);

[C++] public: void WriteDocument(Object\* document, Stream\* stream);

[VB] Overrides Public Sub WriteDocument(ByVal document As Object, ByVal stream As Stream)

[JScript] public override function WriteDocument(document : Object, stream : Stream);

### Description

Writes the passed-in Service Description into the passed-in **System.IO.Stream** . The **System.Web.Services.Description.ServiceDescription**

to write into *stream*. The **System.IO.Stream** into which the serialized **System.Web.Services.Description.ServiceDescription** is written.

ContractSearchPattern class (System.Web.Services.Discovery)

WriteDocument

### *Description*

Represents a contract search pattern. This class cannot be inherited.

ContractSearchPattern

### *Example Syntax:*

WriteDocument

[C#] public ContractSearchPattern();

[C++] public: ContractSearchPattern();

[VB] Public Sub New()

[JScript] public function ContractSearchPattern();

Pattern

WriteDocument

[C#] public override string Pattern {get;}

[C++] public: \_\_property virtual String\* get\_Pattern();

[VB] Overrides Public ReadOnly Property Pattern As String

[JScript] public function get Pattern() : String;

### *Description*

Overrides the base implementation to return the literal string, "\*.asmx".

GetDiscoveryReference

[C#] public override DiscoveryReference GetDiscoveryReference(string filename);

[C++] public: DiscoveryReference\* GetDiscoveryReference(String\* filename);

[VB] Overrides Public Function GetDiscoveryReference(ByVal filename As String) As DiscoveryReference

[JScript] public override function GetDiscoveryReference(filename : String) : DiscoveryReference;

### *Description*

Overrides the base implementation to create a **System.Web.Services.Discovery.ContractReference** object with the specified filename.

**Return Value:** A **System.Web.Services.Discovery.ContractReference** object with the specified filename. A string that represents a filename.

DiscoveryClientDocumentCollection class

(System.Web.Services.Discovery)

ToString

### *Description*

Represents a collection of documents discovered during Web Services discovery that have been downloaded to the client. This class cannot be inherited.

1           The  
2   **System.Web.Services.Discovery.DiscoveryClientProtocol.Documents** property  
3   of **System.Web.Services.Discovery.DiscoveryClientProtocol** is of type  
4   **System.Web.Services.Discovery.DiscoveryClientDocumentCollection** .

5           DiscoveryClientDocumentCollection

6           *Example Syntax:*

7           ToString

9   [C#] public DiscoveryClientDocumentCollection();

10   [C++] public: DiscoveryClientDocumentCollection();

11   [VB] Public Sub New()

12   [JScript] public function DiscoveryClientDocumentCollection();

13           Count

14           Dictionary

15           InnerHashtable

16           Item

17           ToString

20   *Description*

21           Gets or sets a client discovery document object from the  
22   **System.Web.Services.Discovery.DiscoveryClientDocumentCollection** with the  
23   specified URL. The URL of the discovery document to get or set from the  
24   **System.Web.Services.Discovery.DiscoveryClientDocumentCollection**.

25           Keys

ToString

[C#] public ICollection Keys {get;}

[C++] public: \_\_property ICollection\* get\_Keys();

[VB] Public ReadOnly Property Keys As ICollection

[JScript] public function get Keys() : ICollection;

*Description*

Gets an **System.Collections.ICollection** object with all of the keys in the **System.Web.Services.Discovery.DiscoveryClientDocumentCollection** .

Values

ToString

[C#] public ICollection Values {get;}

[C++] public: \_\_property ICollection\* get\_Values();

[VB] Public ReadOnly Property Values As ICollection

[JScript] public function get Values() : ICollection;

*Description*

Gets an **System.Collections.ICollection** object with all of the values in the **System.Web.Services.Discovery.DiscoveryClientDocumentCollection** .

Add

[C#] public void Add(string url, object value);

[C++] public: void Add(String\* url, Object\* value);

1 [VB] Public Sub Add(ByVal url As String, ByVal value As Object)

2 [JScript] public function Add(url : String, value : Object);

3  
4 *Description*

5 Adds an object with the specified URL to the

6 **System.Web.Services.Discovery.DiscoveryClientDocumentCollection** . The

7 URL for the document to add to the

8 **System.Web.Services.Discovery.DiscoveryClientDocumentCollection** . A

9 discovered document to add to the

10 **System.Web.Services.Discovery.DiscoveryClientDocumentCollection** .

11 Contains

12  
13 [C#] public bool Contains(string url);

14 [C++] public: bool Contains(String\* url);

15 [VB] Public Function Contains(ByVal url As String) As Boolean

16 [JScript] public function Contains(url : String) : Boolean;

17  
18 *Description*

19 Determines if the

20 **System.Web.Services.Discovery.DiscoveryClientDocumentCollection** contains

21 an object with the specified URL.

22 *Return Value:* **true** if the

23 **System.Web.Services.Discovery.DiscoveryClientDocumentCollection** contains

24 an object with the specified URL; otherwise, **false** . The URL for the document to

25

locate within the

**System.Web.Services.Discovery.DiscoveryClientDocumentCollection .**

Remove

[C#] public void Remove(string url);

[C++] public: void Remove(String\* url);

[VB] Public Sub Remove(ByVal url As String)

[JScript] public function Remove(url : String);

#### *Description*

Removes an object with the specified URL from the

**System.Web.Services.Discovery.DiscoveryClientDocumentCollection .** The

URL for the discovered document to remove from the

**System.Web.Services.Discovery.DiscoveryClientDocumentCollection.**

DiscoveryClientProtocol class (System.Web.Services.Discovery)

ToString

#### *Description*

Provides support for programmatically invoking Web Services discovery.

Web Service discovery is the process of locating, or discovering, one or more related documents that describe available Web Services. It is through Web Services discovery that Web Service clients learn about the available Web Services at a given URL and how to use them. Web Services discovery works from the premise that you have already obtained the URL to a discovery

document, possibly through a directory service, such as <http://uddi.microsoft.com>, however, you do not have the details about the Web Services offered. Through Web Services discovery, you can discover the details about the Web Services listed in a **System.Web.Services.Discovery.DiscoveryDocument** at a specific URL.

DiscoveryClientProtocol

*Example Syntax:*

ToString

[C#] public DiscoveryClientProtocol();

[C++] public: DiscoveryClientProtocol();

[VB] Public Sub New()

[JScript] public function DiscoveryClientProtocol();

#### *Description*

Initializes a new instance of the **System.Web.Services.Discovery.DiscoveryClientProtocol** class.

AdditionalInformation

ToString

[C#] public IList AdditionalInformation {get;}

[C++] public: \_\_property IList\* get\_AdditionalInformation();

[VB] Public ReadOnly Property AdditionalInformation As IList

[JScript] public function get AdditionalInformation() : IList;



1  
2 *Description*

3 Gets information in addition to references found in the discovery document.

4 The

5 **System.Web.Services.Discovery.DiscoveryClientProtocol.AdditionalInformati**

6 **on** property primarily contains SOAP bindings, represented by the

7 **System.Web.Services.Discovery.SoapBinding** class, defined in the discovery

8 document.

9 AllowAutoRedirect

10 ClientCertificates

11 ConnectionGroupName

12 Container

13 CookieContainer

14 Credentials

15 DesignMode

16 Documents

17 ToString

18  
19  
20 *Description*

21 Gets a collection of discovery documents.

22 The

23 **System.Web.Services.Discovery.DiscoveryClientProtocol.Documents**

24 collection is populated during invocations to the

25 **System.Web.Services.Discovery.DiscoveryClientProtocol.Discover(System.Str**

ing) ,

**System.Web.Services.Discovery.DiscoveryClientProtocol.DiscoverAny(System.String) ,**

**System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveAll** and **System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveOneLevel** methods. During invocations to the **System.Web.Services.Discovery.DiscoveryClientProtocol.Discover(System.String)** and **System.Web.Services.Discovery.DiscoveryClientProtocol.DiscoverAny(System.String)** methods, if the supplied URL is a valid discovery document, that document is added to the **System.Web.Services.Discovery.DiscoveryClientProtocol.Documents** collection. During invocations to the **System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveAll** and **System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveOneLevel** methods, valid discovery document references in the **System.Web.Services.Discovery.DiscoveryClientProtocol.References** collection are added to the **System.Web.Services.Discovery.DiscoveryClientProtocol.Documents** collection.

Errors

ToString

[C#] public DiscoveryExceptionDictionary Errors {get;}

[C++] public: \_\_property DiscoveryExceptionDictionary\* get\_Errors();

1 [VB] Public ReadOnly Property Errors As DiscoveryExceptionDictionary

2 [JScript] public function get Errors() : DiscoveryExceptionDictionary;

3  
4 *Description*

5 Gets a collection of exceptions that occurred during invocation of method  
6 from this class.

7 The **System.Web.Services.Discovery.DiscoveryClientProtocol.Errors**  
8 collection is populated with exceptions that occurred during invocations to the  
9 **System.Web.Services.Discovery.DiscoveryClientProtocol.Discover(System.Str**  
10 **ing)** ,

11 **System.Web.Services.Discovery.DiscoveryClientProtocol.DiscoverAny(Syste**  
12 **m.String)** ,

13 **System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveAll** , and  
14 **System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveOneLevel**  
15 methods. The **System.Web.Services.Discovery.DiscoveryExceptionDictionary**  
16 is cleared on invocation of these methods.

17 Events

18 PreAuthenticate

19 Proxy

20 References

21 ToString

22  
23  
24 *Description*

25 A collection of references founds in resolved discovery documents.

The

**System.Web.Services.Discovery.DiscoveryClientProtocol.References**

collection is populated during invocations to the

**System.Web.Services.Discovery.DiscoveryClientProtocol.Discover(System.String)** ,

**System.Web.Services.Discovery.DiscoveryClientProtocol.DiscoverAny(System.String)** ,

**System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveAll** and

**System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveOneLevel**

methods. During invocations to the

**System.Web.Services.Discovery.DiscoveryClientProtocol.Discover(System.String)** and

**System.Web.Services.Discovery.DiscoveryClientProtocol.DiscoverAny(System.String)** methods, if the supplied URL is a valid discovery document, that

document is added to the

**System.Web.Services.Discovery.DiscoveryClientProtocol.References**

collection together with references found in the discovery document. References added during invocations to the

**System.Web.Services.Discovery.DiscoveryClientProtocol.Discover(System.String)** and

**System.Web.Services.Discovery.DiscoveryClientProtocol.DiscoverAny(System.String)** methods are not necessarily valid discovery documents. During

invocations to

**System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveAll** and

**System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveOneLevel**

references within the

**System.Web.Services.Discovery.DiscoveryClientProtocol.References**

collection are verified as being valid discovery documents. If they are valid and contain references, they are also added to the

**System.Web.Services.Discovery.DiscoveryClientProtocol.References**

collection.

RequestEncoding

Site

Timeout

Url

UserAgent

Discover

[C#] public DiscoveryDocument Discover(string url);

[C++] public: DiscoveryDocument\* Discover(String\* url);

[VB] Public Function Discover(ByVal url As String) As DiscoveryDocument

[JScript] public function Discover(url : String) : DiscoveryDocument;

### *Description*

Discovers the supplied URL to determine if it is a discovery document.

**Return Value:** A **System.Web.Services.Discovery.DiscoveryDocument**

containing the results of Web Services discovery at the supplied URL.

The

**System.Web.Services.Discovery.DiscoveryClientProtocol.Discover(System.Str**

**ing)** method expects that the supplied URL is a discovery document. If the URL

1 refers to a Service Description or an XML schema an exception is thrown. To  
2 discover an XML schema or a Service Description invoke the  
3 **System.Web.Services.Discovery.DiscoveryClientProtocol.DiscoverAny(Syste**  
4 **m.String)** method. The URL where Web Services discovery begins.

#### 5 DiscoverAny

6  
7 [C#] public DiscoveryDocument DiscoverAny(string url);  
8 [C++] public: DiscoveryDocument\* DiscoverAny(String\* url);  
9 [VB] Public Function DiscoverAny(ByVal url As String) As DiscoveryDocument  
10 [JScript] public function DiscoverAny(url : String) : DiscoveryDocument;

#### 12 *Description*

13 Discovers the supplied URL to determine if it is a discovery document,  
14 Service Description or an XML Schema Definition (XSD) schema.

15 *Return Value:* A **System.Web.Services.Discovery.DiscoveryDocument**  
16 containing the results of Web Services discovery at the supplied URL. If the *url*  
17 parameter refers to a Service Description or an XSD Schema, a  
18 **System.Web.Services.Discovery.DiscoveryDocument** is created in memory for  
19 it.

#### 20 The

21 **System.Web.Services.Discovery.DiscoveryClientProtocol.DiscoverAny(Syste**  
22 **m.String)** method discovers whether the supplied URL is a discovery document,  
23 Service Description or XSD schema. If it is known that the URL only refers to a  
24 discovery document, the  
25

**System.Web.Services.Discovery.DiscoveryClientProtocol.Discover(System.String)** method might be invoked. The URL where Web Services discovery begins.

#### Download

```
[C#] public Stream Download(ref string url);  
[C++] public: Stream* Download(String** url);  
[VB] Public Function Download(ByRef url As String) As Stream  
[JScript] public function Download(url : String) : Stream; Downloads the  
discovery document at the supplied URL into a System.IO.Stream object.
```

#### *Description*

Downloads the discovery document at the supplied URL into a **System.IO.Stream** object.

*Return Value:* A **System.IO.Stream** containing the document at the supplied URL. The URL of the discovery document to download.

#### Download

```
[C#] public Stream Download(ref string url, ref string contentType);  
[C++] public: Stream* Download(String** url, String** contentType);  
[VB] Public Function Download(ByRef url As String, ByRef contentType As  
String) As Stream  
[JScript] public function Download(url : String, contentType : String) : Stream;
```

#### *Description*

Downloads the discovery document at the supplied URL into a **System.IO.Stream** object, setting the *contentType* parameter to the MIME encoding of the discovery document.

*Return Value:* A **System.IO.Stream** containing the document at the supplied URL. The URL of the discovery document to download. The MIME encoding of the downloaded discovery document.

## ReadAll

```
[C#] public DiscoveryClientResultCollection ReadAll(string topLevelFilename);
[C++] public: DiscoveryClientResultCollection* ReadAll(String*
topLevelFilename);
[VB] Public Function ReadAll(ByVal topLevelFilename As String) As
DiscoveryClientResultCollection
[JScript] public function ReadAll(topLevelFilename : String) :
DiscoveryClientResultCollection;
```

## Description

Reads in a file containing a map of saved discovery documents populating the **System.Web.Services.Discovery.DiscoveryClientProtocol.Documents** and **System.Web.Services.Discovery.DiscoveryClientProtocol.References** properties, with discovery documents, XML Schema Definition (XSD) schemas, and Service Descriptions referenced in the file.

*Return Value:* A

**System.Web.Services.Discovery.DiscoveryClientResultCollection** containing the results found in the file with the map of saved discovery documents. The file



format is a

**System.Web.Services.Discovery.DiscoveryClientProtocol.DiscoveryClientResultsFile** class serialized into XML; however, one would typically create the file using only the **System.Web.Services.Discovery.DiscoveryClientProtocol.WriteAll(System.String,System.String)** method or Disco.exe.

A file containing a map of saved discovery documents can be created by the **System.Web.Services.Discovery.DiscoveryClientProtocol.WriteAll(System.String,System.String)** method or Disco.exe. Name of file to read in, containing the map of saved discovery documents.

**ResolveAll**

[C#] public void ResolveAll();

[C++] public: void ResolveAll();

[VB] Public Sub ResolveAll()

[JScript] public function ResolveAll();

### *Description*

Resolves all references to discovery documents, XML Schema Definition (XSD) schemas, and Service Descriptions in the **System.Web.Services.Discovery.DiscoveryClientProtocol.References** property, as well as references found in referring discovery documents.

**System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveAll** resolves all valid references it finds and places them in the **System.Web.Services.Discovery.DiscoveryClientProtocol.Documents** property.

Both **System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveAll** and **System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveOneLevel** download and verify XSD schemas and Service Descriptions in the **System.Web.Services.Discovery.DiscoveryClientProtocol.References** property. However, the two methods differ in how they handle discovery documents.

#### **ResolveOneLevel**

[C#] public void ResolveOneLevel();

[C++] public: void ResolveOneLevel();

[VB] Public Sub ResolveOneLevel()

[JScript] public function ResolveOneLevel();

#### *Description*

Resolves all references to discovery documents, XML Schema Definition (XSD) schemas and Service Descriptions in **System.Web.Services.Discovery.DiscoveryClientProtocol.References**, as well as references found in those discovery documents.

**System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveOneLevel** resolves all valid references it finds and places them in the **System.Web.Services.Discovery.DiscoveryClientProtocol.Documents** property. Both **System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveAll** and **System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveOneLevel** download and verify XSD schemas and Service Descriptions in the **System.Web.Services.Discovery.DiscoveryClientProtocol.References** property. However, the two methods differ in how they handle discovery documents.

## WriteAll

```
[C#] public DiscoveryClientResultCollection WriteAll(string directory, string
topLevelFilename);

[C++] public: DiscoveryClientResultCollection* WriteAll(String* directory,
String* topLevelFilename);

[VB] Public Function WriteAll(ByVal directory As String, ByVal
topLevelFilename As String) As DiscoveryClientResultCollection

[JScript] public function WriteAll(directory : String, topLevelFilename : String) :
DiscoveryClientResultCollection;
```

### *Description*

Writes all discovery documents, XML Schema Definition (XSD) schemas, and Service Descriptions in the **System.Web.Services.Discovery.DiscoveryClientProtocol.Documents** property to the supplied directory and creates a file in that directory.

### *Return Value:* A

**System.Web.Services.Discovery.DiscoveryClientResultCollection** containing the results of all files saved.

The file created with the name of the *topLevelFilename* parameter in the directory specified by the *directory* parameter contains a map of saved discovery documents, XML Schema Definition (XSD) schemas and Service Descriptions.

This file can be read in using the

**System.Web.Services.Discovery.DiscoveryClientProtocol.ReadAll(System.String)** method to populate the

1 **System.Web.Services.Discovery.DiscoveryClientProtocol.References** and  
2 **System.Web.Services.Discovery.DiscoveryClientProtocol.Documents**  
3 properties. The format of the file is XML containing an serialized version of the  
4 **System.Web.Services.Discovery.DiscoveryClientProtocol.DiscoveryClientRes**  
5 **ultsFile** class. The directory in which to save all documents currently in the  
6 **System.Web.Services.Discovery.DiscoveryClientProtocol.Documents** property.  
7 The name of the file to create or overwrite containing a map of all documents  
8 saved.

9       DiscoveryClientReferenceCollection class  
10 (System.Web.Services.Discovery)

11       WriteAll

12  
13  
14 *Description*

15       Represents a collection of  
16 **System.Web.Services.Discovery.DiscoveryReference** objects. This class cannot  
17 be inherited.

18       The  
19 **System.Web.Services.Discovery.DiscoveryClientProtocol.References** property  
20 of **System.Web.Services.Discovery.DiscoveryClientProtocol** is of type  
21 **System.Web.Services.Discovery.DiscoveryClientReferenceCollection** .

22       DiscoveryClientReferenceCollection

23 *Example Syntax:*

24       WriteAll  
25

```

1
2 [C#] public DiscoveryClientReferenceCollection();
3 [C++] public: DiscoveryClientReferenceCollection();
4 [VB] Public Sub New()
5 [JScript] public function DiscoveryClientReferenceCollection();

```

Count

Dictionary

InnerHashtable

Item

WriteAll

### *Description*

Gets or sets a **System.Web.Services.Discovery.DiscoveryReference** object from the **System.Web.Services.Discovery.DiscoveryClientReferenceCollection** with the specified URL. The URL for the DiscoveryReference to get or set from the **System.Web.Services.Discovery.DiscoveryClientReferenceCollection**.

Keys

WriteAll

```

22 [C#] public ICollection Keys {get;}
23 [C++] public: __property ICollection* get_Keys();
24 [VB] Public ReadOnly Property Keys As ICollection
25 [JScript] public function get Keys() : ICollection;

```

1  
2 *Description*

3 Gets an **System.Collections.ICollection** object with all of the keys in the  
4 **System.Web.Services.Discovery.DiscoveryClientReferenceCollection** .

5 Values

6 WriteAll

7  
8 [C#] public ICollection Values {get;}

9 [C++] public: \_\_property ICollection\* get\_Values();

10 [VB] Public ReadOnly Property Values As ICollection

11 [JScript] public function get Values() : ICollection;

12  
13 *Description*

14 Gets an **System.Collections.ICollection** object with all of the values in the  
15 **System.Web.Services.Discovery.DiscoveryClientReferenceCollection** .

16 Add

17  
18 [C#] public void Add(DiscoveryReference value);

19 [C++] public: void Add(DiscoveryReference\* value);

20 [VB] Public Sub Add(ByVal value As DiscoveryReference)

21 [JScript] public function Add(value : DiscoveryReference); Adds a

22 **System.Web.Services.Discovery.DiscoveryReference** to the

23 **System.Web.Services.Discovery.DiscoveryClientReferenceCollection** .

24  
25 *Description*

1 Adds a **System.Web.Services.Discovery.DiscoveryReference** to the  
2 **System.Web.Services.Discovery.DiscoveryClientReferenceCollection** . The  
3 **System.Web.Services.Discovery.DiscoveryReference** to add to the  
4 **System.Web.Services.Discovery.DiscoveryClientReferenceCollection**.

5 Add

6  
7 [C#] public void Add(string url, DiscoveryReference value);

8 [C++] public: void Add(String\* url, DiscoveryReference\* value);

9 [VB] Public Sub Add(ByVal url As String, ByVal value As DiscoveryReference)

10 [JScript] public function Add(url : String, value : DiscoveryReference);

11  
12 *Description*

13 Adds a **System.Web.Services.Discovery.DiscoveryReference** with the  
14 specified URL and value to the

15 **System.Web.Services.Discovery.DiscoveryClientReferenceCollection** . The  
16 URL for the reference to add to the

17 **System.Web.Services.Discovery.DiscoveryClientReferenceCollection**. The  
18 **DiscoveryReference** to add to the

19 **System.Web.Services.Discovery.DiscoveryClientReferenceCollection**.

20 Contains

21  
22 [C#] public bool Contains(string url);

23 [C++] public: bool Contains(String\* url);

24 [VB] Public Function Contains(ByVal url As String) As Boolean

25 [JScript] public function Contains(url : String) : Boolean;

1  
2 *Description*

3       Determines if the  
4 **System.Web.Services.Discovery.DiscoveryClientReferenceCollection** contains  
5 a **System.Web.Services.Discovery.DiscoveryReference** with the specified URL.

6 *Return Value:* **true** if the  
7 **System.Web.Services.Discovery.DiscoveryClientReferenceCollection** contains  
8 a **System.Web.Services.Discovery.DiscoveryReference** with the specified URL;  
9 otherwise, **false** . The URL for the  
10 **System.Web.Services.Discovery.DiscoveryReference** to locate within the  
11 **System.Web.Services.Discovery.DiscoveryClientReferenceCollection**.

12       Remove

13  
14 [C#] public void Remove(string url);  
15 [C++] public: void Remove(String\* url);  
16 [VB] Public Sub Remove(ByVal url As String)  
17 [JScript] public function Remove(url : String);  
18

19 *Description*

20       Removes a **System.Web.Services.Discovery.DiscoveryReference** with the  
21 specified URL from the  
22 **System.Web.Services.Discovery.DiscoveryClientReferenceCollection** . A string  
23 that represents the URL for the object to remove from the  
24 **System.Web.Services.Discovery.DiscoveryClientReferenceCollection**.

25       DiscoveryClientResult class (System.Web.Services.Discovery)



ToString

*Description*

Represents the details of a discovery reference without the actual contents of the referenced document. This class cannot be inherited.

Using the

**System.Web.Services.Discovery.DiscoveryClientProtocol.WriteAll(System.String, System.String)** method of

**System.Web.Services.Discovery.DiscoveryClientProtocol**, all valid referenced documents and a file containing a map of all save documents can be written to disk. The file containing a map of all saved documents contains the details of each document as summarized by the properties of

**System.Web.Services.Discovery.DiscoveryClientResult**.

DiscoveryClientResult

*Example Syntax:*

ToString

[C#] public DiscoveryClientResult();

[C++] public: DiscoveryClientResult();

[VB] Public Sub New()

[JScript] public function DiscoveryClientResult(); Initializes a new instance of the **System.Web.Services.Discovery.DiscoveryClientResult** class.

*Description*

1        Initializes a new instance of the  
2        **System.Web.Services.Discovery.DiscoveryClientResult** class.

3        **DiscoveryClientResult**

4        *Example Syntax:*

5        **ToString**

6  
7        [C#] public DiscoveryClientResult(Type referenceType, string url, string  
8        filename);

9        [C++] public: DiscoveryClientResult(Type\* referenceType, String\* url, String\*  
10        filename);

11        [VB] Public Sub New(ByVal referenceType As Type, ByVal url As String, ByVal  
12        filename As String)

13        [JScript] public function DiscoveryClientResult(referenceType : Type, url : String,  
14        filename : String);

15  
16        *Description*

17        Initializes a new instance of the  
18        **System.Web.Services.Discovery.DiscoveryClientResult** class and sets the  
19        **System.Web.Services.Discovery.DiscoveryClientResult.ReferenceTypeName**  
20        property to *referenceType* , the

21        **System.Web.Services.Discovery.DiscoveryClientResult.Url** property to *url* and  
22        the **System.Web.Services.Discovery.DiscoveryClientResult.Filename** property  
23        to *filename* . Name of the type for a reference in the discovery document. Sets the

24        **System.Web.Services.Discovery.DiscoveryClientResult.ReferenceTypeName**  
25        property. URL for the reference. Sets the

1 **System.Web.Services.Discovery.DiscoveryClientResult.Url** property. Name of  
2 the file in which the reference was saved. Sets the

3 **System.Web.Services.Discovery.DiscoveryClientResult.Filename** property.

4       Filename

5       ToString

6  
7 [C#] public string Filename {get; set;}

8 [C++] public: \_\_property String\* get\_Filename();public: \_\_property void  
9 set\_Filename(String\*);

10 [VB] Public Property Filename As String

11 [JScript] public function get Filename() : String;public function set  
12 Filename(String);

13  
14 *Description*

15       Gets or sets the name of the file in which the reference is saved.

16       The name of the file is the name of the file saved on the local machine  
17 based on the contents of the file. For instance, if the referenced file is a discovery  
18 document, the saved file will have a .disco extension on a Windows operating  
19 system.

20       ReferenceTypeName

21       ToString

22  
23 [C#] public string ReferenceTypeName {get; set;}

24 [C++] public: \_\_property String\* get\_ReferenceTypeName();public: \_\_property  
25 void set\_ReferenceTypeName(String\*);

1 [VB] Public Property ReferenceTypeName As String

2 [JScript] public function get ReferenceTypeName() : String;public function set

3 ReferenceTypeName(String);

4  
5 *Description*

6 Name of the type for a reference in the discovery document.

7 A reference within a discovery document can contain references to Service  
8 Descriptions, XSD schemas, or other discovery documents. Therefore,

9 **System.Web.Services.Discovery.DiscoveryClientResult.ReferenceTypeName**

10 can have the following values:

11 **System.Web.Services.Description.ServiceDescription** ,

12 **System.Xml.Schema.XmlSchema** , and

13 **System.Web.Services.Discovery.DiscoveryDocument** .

14 Url

15 ToString

16  
17 [C#] public string Url {get; set;}

18 [C++] public: \_\_property String\* get\_Url();public: \_\_property void

19 set\_Url(String\*);

20 [VB] Public Property Url As String

21 [JScript] public function get Url() : String;public function set Url(String);

22  
23 *Description*

24 Gets or sets the URL for the reference.

The URL can refer to a discovery document, XML Schema Definition (XSD) schema, or a Service Description.

DiscoveryClientResultCollection class (System.Web.Services.Discovery)

ToString

### *Description*

Contains a collection of **System.Web.Services.Discovery.DiscoveryClientResult** objects. This class cannot be inherited.

DiscoveryClientResultCollection

### *Example Syntax:*

ToString

[C#] public DiscoveryClientResultCollection();

[C++] public: DiscoveryClientResultCollection();

[VB] Public Sub New()

[JScript] public function DiscoveryClientResultCollection();

Count

InnerList

Item

ToString

### *Description*

Gets or sets the **System.Web.Services.Discovery.DiscoveryClientResult** at position *i* of the

**System.Web.Services.Discovery.DiscoveryClientResultCollection** .

This property provides the ability to access a specific element in the collection by using the following syntax: myCollection[index] . The zero-based index of the **System.Web.Services.Discovery.DiscoveryClientResult** to get or set.

List

Add

[C#] public int Add(DiscoveryClientResult value);

[C++] public: int Add(DiscoveryClientResult\* value);

[VB] Public Function Add(ByVal value As DiscoveryClientResult) As Integer

[JScript] public function Add(value : DiscoveryClientResult) : int;

#### *Description*

Adds a **System.Web.Services.Discovery.DiscoveryClientResult** to the **System.Web.Services.Discovery.DiscoveryClientResultCollection** .

*Return Value:* The position into which the **System.Web.Services.Discovery.DiscoveryClientResult** was inserted. The **System.Web.Services.Discovery.DiscoveryClientResult** to add to the **System.Web.Services.Discovery.DiscoveryClientResultCollection**.

Contains

[C#] public bool Contains(DiscoveryClientResult value);

1 [C++] public: bool Contains(DiscoveryClientResult\* value);

2 [VB] Public Function Contains(ByVal value As DiscoveryClientResult) As

3 Boolean

4 [JScript] public function Contains(value : DiscoveryClientResult) : Boolean;

5  
6 *Description*

7 Determines whether the

8 **System.Web.Services.Discovery.DiscoveryClientResultCollection** contains a  
9 specific **System.Web.Services.Discovery.DiscoveryClientResult** .

10 *Return Value:* **true** if the

11 **System.Web.Services.Discovery.DiscoveryClientResult** is found in the  
12 **System.Web.Services.Discovery.DiscoveryClientResultCollection** ; otherwise,  
13 **false** . The **System.Web.Services.Discovery.DiscoveryClientResult** to locate in  
14 the **System.Web.Services.Discovery.DiscoveryClientResultCollection**.

15 Remove

16  
17 [C#] public void Remove(DiscoveryClientResult value);

18 [C++] public: void Remove(DiscoveryClientResult\* value);

19 [VB] Public Sub Remove(ByVal value As DiscoveryClientResult)

20 [JScript] public function Remove(value : DiscoveryClientResult);

21  
22 *Description*

23 Removes the first occurrence of a specific

24 **System.Web.Services.Discovery.DiscoveryClientResult** from the

25 **System.Web.Services.Discovery.DiscoveryClientResultCollection** .

1 The **System.Web.Services.Discovery.DiscoveryClientResult** elements  
2 that follow the removed **System.Web.Services.Discovery.DiscoveryClientResult**  
3 move up to occupy the vacated spot. The  
4 **System.Web.Services.Discovery.DiscoveryClientResult** to remove from the  
5 **System.Web.Services.Discovery.DiscoveryClientResultCollection**.

6 **DiscoveryClientProtocol.DiscoveryClientResultsFile** class  
7 (**System.Web.Services.Discovery**)

8 **ToString**

9  
10  
11 *Description*

12 Represents the root element of an XML document containing the results of  
13 all files written when the  
14 **System.Web.Services.Discovery.DiscoveryClientProtocol.WriteAll(System.Str**  
15 **ing,System.String)** method is invoked.

16 When you invoke the  
17 **System.Web.Services.Discovery.DiscoveryClientProtocol.WriteAll(System.Str**  
18 **ing,System.String)** method, all resolved discovery documents and a file  
19 containing a map of all those files are saved to a directory. The map file is  
20 described in XML with the root element being  
21 **System.Web.Services.Discovery.DiscoveryClientProtocol.DiscoveryClientRes**  
22 **ultsFile** ; this class is passed to the **System.Xml.Serialization.XmlSerializer**  
23 class to serialize the results.

24 **DiscoveryClientProtocol.DiscoveryClientResultsFile**

25 *Example Syntax:*



ToString

[C#] public DiscoveryClientProtocol.DiscoveryClientResultsFile();

[C++] public: DiscoveryClientResultsFile();

[VB] Public Sub New()

[JScript] public function DiscoveryClientProtocol.DiscoveryClientResultsFile();

Results

ToString

[C#] public DiscoveryClientResultCollection Results {get;}

[C++] public: \_\_property DiscoveryClientResultCollection\* get\_Results();

[VB] Public ReadOnly Property Results As DiscoveryClientResultCollection

[JScript] public function get Results() : DiscoveryClientResultCollection;

### *Description*

Gets a collection of

**System.Web.Services.Discovery.DiscoveryClientResult** objects.

DiscoveryDocument class (System.Web.Services.Discovery)

ToString

### *Description*

Represents a discovery document. This class cannot be inherited.

Web Services discovery involves discovering the available Web Services given an URL. The URL typically points to a discovery document, which usually

has a .disco file extension. The discovery document, which is an XML document, contains references to information about the existence of Web Services, such as a Service Description, XML Schema Definition language (XSD) schema, or another discovery document. This class represents the contents of the discovery document; where the **System.Web.Services.Discovery.DiscoveryDocument.References** property contains a list of the references contained within the discovery document.

ToString

[C#] public const string Namespace;

[C++] public: const String\* Namespace;

[VB] Public Const Namespace As String

[JScript] public var Namespace : String;

#### *Description*

Namespace of the discovery XML element of a discovery document.

A discovery document contains references to documents describing Web Services. These references are XML elements contained within a discovery XML element. That discovery XML element is a member of the XML namespace specified by the

**System.Web.Services.Discovery.DiscoveryDocument.Namespace** constant.

DiscoveryDocument

#### *Example Syntax:*

ToString

[C#] public DiscoveryDocument();

1 [C++] public: DiscoveryDocument();  
2 [VB] Public Sub New()  
3 [JScript] public function DiscoveryDocument();  
4

5 *Description*

6       Initializes a new instance of the  
7 **System.Web.Services.Discovery.DiscoveryDocument** class.

8       References

9       ToString  
10

11 [C#] public IList References {get;}

12 [C++] public: \_\_property IList\* get\_References();

13 [VB] Public ReadOnly Property References As IList

14 [JScript] public function get References() : IList;  
15

16 *Description*

17       A list of references contained within the discovery document.

18       A discovery document contains references to information about the  
19 existence of Web Services. These references can refer to Service Descriptions,  
20 XSD schemas, or other discovery documents. The  
21 **System.Web.Services.Discovery.DiscoveryDocument.References** property  
22 contains a list of those references.

23       CanRead  
24

25 [C#] public static bool CanRead(XmlReader xmlReader);

1 [C++] public: static bool CanRead(XmlReader\* xmlReader);

2 [VB] Public Shared Function CanRead(ByVal xmlReader As XmlReader) As

3 Boolean

4 [JScript] public static function CanRead(xmlReader : XmlReader) : Boolean;

6 *Description*

7 Returns a value indicating whether the passed **System.Xml.XmlReader**  
8 can be deserialized into a **System.Web.Services.Discovery.DiscoveryDocument**

10 *Return Value:* **true** if **System.Xml.XmlReader** can be deserialized into a  
11 **System.Web.Services.Discovery.DiscoveryDocument** ; otherwise, **false** . The  
12 **System.Xml.XmlReader** to determine whether it can be deserialized into a  
13 **System.Web.Services.Discovery.DiscoveryDocument**.

14 *Read*

16 [C#] public static DiscoveryDocument Read(Stream stream);

17 [C++] public: static DiscoveryDocument\* Read(Stream\* stream);

18 [VB] Public Shared Function Read(ByVal stream As Stream) As

19 DiscoveryDocument

20 [JScript] public static function Read(stream : Stream) : DiscoveryDocument;

21 Reads and returns a **System.Web.Services.Discovery.DiscoveryDocument** from  
22 the passed object.

24 *Description*

1 Reads and returns a  
2 **System.Web.Services.Discovery.DiscoveryDocument** from the passed  
3 **System.IO.Stream** .

4 *Return Value:* A **System.Web.Services.Discovery.DiscoveryDocument**  
5 containing the contents of a discovery document from the passed  
6 **System.IO.Stream** . The **System.IO.Stream** from which to read the  
7 **System.Web.Services.Discovery.DiscoveryDocument** .

8 Read

9  
10 [C#] public static DiscoveryDocument Read(Reader reader);  
11 [C++] public: static DiscoveryDocument\* Read(Reader\* reader);  
12 [VB] Public Shared Function Read(ByVal reader As Reader) As  
13 DiscoveryDocument  
14 [JScript] public static function Read(reader : Reader) : DiscoveryDocument;

15  
16 *Description*

17 Reads and returns a  
18 **System.Web.Services.Discovery.DiscoveryDocument** from the passed  
19 **System.IO.TextReader** .  
20 *Return Value:* A **System.Web.Services.Discovery.DiscoveryDocument**  
21 containing the contents of a discovery document from the passed  
22 **System.IO.TextReader** . The **System.IO.TextReader** from which to read the  
23 **System.Web.Services.Discovery.DiscoveryDocument** .

24 Read  
25

```

1
2 [C#] public static DiscoveryDocument Read(XmlReader xmlReader);
3 [C++] public: static DiscoveryDocument* Read(XmlReader* xmlReader);
4 [VB] Public Shared Function Read(ByVal xmlReader As XmlReader) As
5 DiscoveryDocument
6 [JScript] public static function Read(xmlReader : XmlReader) :
7 DiscoveryDocument;
8

```

### *Description*

Reads and returns a **System.Web.Services.Discovery.DiscoveryDocument** from the passed **System.Xml.XmlReader** .

*Return Value:* A **System.Web.Services.Discovery.DiscoveryDocument** containing the contents of a discovery document from the passed **System.Xml.XmlReader** . The **System.Xml.XmlReader** from which to read the **System.Web.Services.Discovery.DiscoveryDocument** .

Write

```

18
19 [C#] public void Write(Stream stream);
20 [C++] public: void Write(Stream* stream);
21 [VB] Public Sub Write(ByVal stream As Stream)
22 [JScript] public function Write(stream : Stream);
23

```

### *Description*

Writes this **System.Web.Services.Discovery.DiscoveryDocument** into the passed **System.IO.Stream** . The **System.IO.Stream** into which this **System.Web.Services.Discovery.DiscoveryDocument** is written.

Write

[C#] public void Write(TextWriter writer);

[C++] public: void Write(TextWriter\* writer);

[VB] Public Sub Write(ByVal writer As TextWriter)

[JScript] public function Write(writer : TextWriter); Writes this

**System.Web.Services.Discovery.DiscoveryDocument** into the passed object.

#### *Description*

Writes this **System.Web.Services.Discovery.DiscoveryDocument** into the passed **System.IO.TextWriter** . The **System.IO.TextWriter** into which this **System.Web.Services.Discovery.DiscoveryDocument** is written.

Write

[C#] public void Write(XmlWriter writer);

[C++] public: void Write(XmlWriter\* writer);

[VB] Public Sub Write(ByVal writer As XmlWriter)

[JScript] public function Write(writer : XmlWriter);

#### *Description*

Writes this **System.Web.Services.Discovery.DiscoveryDocument** into the  
passed **System.Xml.XmlWriter** . The **System.Xml.XmlWriter** into which this  
**System.Web.Services.Discovery.DiscoveryDocument** is written.

DiscoveryDocumentLinksPattern class (System.Web.Services.Discovery)  
Write

### *Description*

DiscoveryDocumentLinksPattern

### *Example Syntax:*

Write

[C#] public DiscoveryDocumentLinksPattern();

[C++] public: DiscoveryDocumentLinksPattern();

[VB] Public Sub New()

[JScript] public function DiscoveryDocumentLinksPattern();

Pattern

Write

[C#] public override string Pattern {get;}

[C++] public: \_\_property virtual String\* get\_Pattern();

[VB] Overrides Public ReadOnly Property Pattern As String

[JScript] public function get Pattern() : String;



1  
2 *Description*

3  
4       GetDiscoveryReference

5  
6 [C#] public override DiscoveryReference GetDiscoveryReference(string  
7 filename);

8 [C++] public: DiscoveryReference\* GetDiscoveryReference(String\* filename);

9 [VB] Overrides Public Function GetDiscoveryReference(ByVal filename As  
10 String) As DiscoveryReference

11 [JScript] public override function GetDiscoveryReference(filename : String) :  
12 DiscoveryReference;

13  
14 *Description*

15  
16       DiscoveryDocumentReference class (System.Web.Services.Discovery)  
17       ToString

18  
19  
20 *Description*

21       Represents a reference to a discovery document. This class cannot be  
22 inherited.

23       Web Services discovery involves discovering the available Web Services  
24 given an URL. The URL typically points to a discovery document, which usually  
25 has a .disco file extension. The discovery document contains references to

1 information about the existence of Web Services, such as Service Descriptions,  
2 XML Schema Definition language (XSD) schemas, or other discovery documents.

3 This class represents a reference to a discovery document.

4     DiscoveryDocumentReference

5     *Example Syntax:*

6     ToString

7  
8 [C#] public DiscoveryDocumentReference();

9 [C++] public: DiscoveryDocumentReference();

10 [VB] Public Sub New()

11 [JScript] public function DiscoveryDocumentReference(); Initializes a new  
12 instance of the **System.Web.Services.Discovery.DiscoveryDocumentReference**  
13 class.

14  
15 *Description*

16     Initializes a new instance of the  
17 **System.Web.Services.Discovery.DiscoveryDocumentReference** class.

18     DiscoveryDocumentReference

19     *Example Syntax:*

20     ToString

21  
22 [C#] public DiscoveryDocumentReference(string href);

23 [C++] public: DiscoveryDocumentReference(String\* href);

24 [VB] Public Sub New(ByVal href As String)

25 [JScript] public function DiscoveryDocumentReference(href : String);

1  
2 *Description*

3       Initializes a new instance of the  
4 **System.Web.Services.Discovery.DiscoveryDocumentReference** class, setting  
5 the **System.Web.Services.Discovery.DiscoveryDocumentReference.Ref**  
6 property to *href*. Reference to a discovery document.

7       ClientProtocol

8       DefaultFilename

9       ToString

10  
11  
12 *Description*

13       Gets the name of the default file to use when saving the referenced  
14 discovery document.

15       Document

16       ToString

17  
18 [C#] public DiscoveryDocument Document {get;}

19 [C++] public: \_\_property DiscoveryDocument\* get\_Document();

20 [VB] Public ReadOnly Property Document As DiscoveryDocument

21 [JScript] public function get Document() : DiscoveryDocument;

22  
23 *Description*

24       Gets the contents of the referenced discovery document as a  
25 **System.Web.Services.Discovery.DiscoveryDocument** object.

If the discovery document has not been downloaded and added to the **System.Web.Services.Discovery.DiscoveryClientProtocol.Documents** property of **System.Web.Services.Discovery.DiscoveryReference.ClientProtocol**, an attempt to download and resolve the document is made.

Ref

ToString

[C#] public string Ref {get; set;}

[C++] public: \_\_property String\* get\_Ref();public: \_\_property void set\_Ref(String\*);

[VB] Public Property Ref As String

[JScript] public function get Ref() : String;public function set Ref(String);

#### *Description*

Gets or sets the reference to a discovery document.

For most references, the

**System.Web.Services.Discovery.DiscoveryDocumentReference.Ref** property is a URL, such as <http://www.contoso.com/MyWebService.disco>.

Url

ToString

[C#] public override string Url {get; set;}

[C++] public: \_\_property virtual String\* get\_Url();public: \_\_property virtual void set\_Url(String\*);

[VB] Overrides Public Property Url As String

1 [JScript] public function getUrl() : String;public function set Url(String);

3 *Description*

4 Gets or sets the URL of the referenced discovery document.

5 Returns the value of the

6 **System.Web.Services.Discovery.DiscoveryDocumentReference.Ref** property.

7 ReadDocument

9 [C#] public override object ReadDocument(Stream stream);

10 [C++] public: Object\* ReadDocument(Stream\* stream);

11 [VB] Overrides Public Function ReadDocument(ByVal stream As Stream) As

12 Object

13 [JScript] public override function ReadDocument(stream : Stream) : Object;

15 *Description*

16 Reads and returns the discovery document from the passed

17 **System.IO.Stream** .

18 *Return Value:* A **System.Web.Services.Discovery.DiscoveryDocument**

19 containing the contents of the referenced discovery document. **System.IO.Stream**

20 containing the discovery document.

21 Resolve

23 [C#] protected internal override void Resolve(string contentType, Stream stream);

24 [C++] protected public: void Resolve(String\* contentType, Stream\* stream);

25 [VB] Overrides Protected Friend Dim Sub Resolve(ByVal contentType As String,

ByVal stream As Stream)

[JScript] package override function Resolve(contentType : String, stream : Stream);

#### *Description*

Resolves whether the referenced document is valid.

If the MIME type is text/xml and the contents of *stream* is a discovery document, then the contents of *stream* are added to the

**System.Web.Services.Discovery.DiscoveryClientProtocol.References** and **System.Web.Services.Discovery.DiscoveryClientProtocol.Documents** properties of

**System.Web.Services.Discovery.DiscoveryReference.ClientProtocol**. The MIME type of *stream*. The **System.IO.Stream** containing the referenced document.

#### *ResolveAll*

[C#] public void ResolveAll();

[C++] public: void ResolveAll();

[VB] Public Sub ResolveAll()

[JScript] public function ResolveAll();

#### *Description*

Verifies that all referenced documents within the discovery document are valid.

Unless you specifically need to resolve the individual references of a **System.Web.Services.Discovery.DiscoveryDocumentReference** object, you should invoke the **System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveAll** or **System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveOneLevel** methods of **System.Web.Services.Discovery.DiscoveryClientProtocol**.

**WriteDocument**

[C#] public override void WriteDocument(object document, Stream stream);  
[C++] public: void WriteDocument(Object\* document, Stream\* stream);  
[VB] Overrides Public Sub WriteDocument(ByVal document As Object, ByVal stream As Stream)  
[JScript] public override function WriteDocument(document : Object, stream : Stream);

#### *Description*

Writes the passed **System.Web.Services.Discovery.DiscoveryDocument** into the passed **System.IO.Stream**. The **System.Web.Services.Discovery.DiscoveryDocument** to write into *stream*. The **System.IO.Stream** into which the serialized discovery document is written.

DiscoveryDocumentSearchPattern class (System.Web.Services.Discovery)

**WriteDocument**

#### *Description*

DiscoveryDocumentSearchPattern

*Example Syntax:*

WriteDocument

[C#] public DiscoveryDocumentSearchPattern();

[C++] public: DiscoveryDocumentSearchPattern();

[VB] Public Sub New()

[JScript] public function DiscoveryDocumentSearchPattern();

Pattern

WriteDocument

[C#] public override string Pattern {get;}

[C++] public: \_\_property virtual String\* get\_Pattern();

[VB] Overrides Public ReadOnly Property Pattern As String

[JScript] public function get Pattern() : String;

*Description*

GetDiscoveryReference

[C#] public override DiscoveryReference GetDiscoveryReference(string filename);

[C++] public: DiscoveryReference\* GetDiscoveryReference(String\* filename);

[VB] Overrides Public Function GetDiscoveryReference(ByVal filename As



String) As DiscoveryReference

[JScript] public override function GetDiscoveryReference(filename : String) :

DiscoveryReference;

*Description*

DiscoveryExceptionDictionary class (System.Web.Services.Discovery)

ToString

*Description*

Collects exceptions that occurred during Web Services discovery. This class cannot be inherited.

The **System.Web.Services.Discovery.DiscoveryClientProtocol.Errors** property of **System.Web.Services.Discovery.DiscoveryClientProtocol** is of type **System.Web.Services.Discovery.DiscoveryExceptionDictionary** .

DiscoveryExceptionDictionary

*Example Syntax:*

ToString

[C#] public DiscoveryExceptionDictionary();

[C++] public: DiscoveryExceptionDictionary();

[VB] Public Sub New()

[JScript] public function DiscoveryExceptionDictionary();

Count

Dictionary

InnerHashtable

Item

ToString

*Description*

Gets or sets the **System.Exception** that occurred while discovering the specified URL from the **System.Web.Services.Discovery.DiscoveryExceptionDictionary** . The URL of the discovery document that caused an exception to be thrown during Web Services discovery.

Keys

ToString

[C#] public ICollection Keys {get;}

[C++] public: \_\_property ICollection\* get\_Keys();

[VB] Public ReadOnly Property Keys As ICollection

[JScript] public function get Keys() : ICollection;

*Description*

Gets a **System.Collections.ICollection** object with all of the keys in the **System.Web.Services.Discovery.DiscoveryExceptionDictionary** .

Values

ToString

1  
2 [C#] public ICollection Values {get;}

3 [C++] public: \_\_property ICollection\* get\_Values();

4 [VB] Public ReadOnly Property Values As ICollection

5 [JScript] public function get Values() : ICollection;

6  
7 *Description*

8 Gets a **System.Collections.ICollection** object containing all of the values  
9 in the **System.Web.Services.Discovery.DiscoveryExceptionDictionary** .

10 Add

11  
12 [C#] public void Add(string url, Exception value);

13 [C++] public: void Add(String\* url, Exception\* value);

14 [VB] Public Sub Add(ByVal url As String, ByVal value As Exception)

15 [JScript] public function Add(url : String, value : Exception);

16  
17 *Description*

18 Adds an **System.Exception** with a key of *url* to the  
19 **System.Web.Services.Discovery.DiscoveryExceptionDictionary** . The URL that  
20 caused an exception during Web Services discovery. The **System.Exception** that  
21 occurred during Web Services discovery.

22 Contains

23  
24 [C#] public bool Contains(string url);

25 [C++] public: bool Contains(String\* url);

1 [VB] Public Function Contains(ByVal url As String) As Boolean

2 [JScript] public function Contains(url : String) : Boolean;

3  
4 *Description*

5 Determines whether the  
6 **System.Web.Services.Discovery.DiscoveryExceptionDictionary** contains an  
7 **System.Exception** with the specified URL.

8 *Return Value:* **true** if the  
9 **System.Web.Services.Discovery.DiscoveryExceptionDictionary** contains an  
10 **System.Exception** with the specified URL; otherwise, **false** . The URL of the  
11 **System.Exception** to locate within the  
12 **System.Web.Services.Discovery.DiscoveryExceptionDictionary**.

13 Remove

14  
15 [C#] public void Remove(string url);

16 [C++] public: void Remove(String\* url);

17 [VB] Public Sub Remove(ByVal url As String)

18 [JScript] public function Remove(url : String);

19  
20 *Description*

21 Removes an **System.Exception** with the specified URL from the  
22 **System.Web.Services.Discovery.DiscoveryExceptionDictionary** . The URL of  
23 the **System.Exception** to remove from the  
24 **System.Web.Services.Discovery.DiscoveryExceptionDictionary**.

25 DiscoveryReference class (System.Web.Services.Discovery)

ToString

*Description*

The base class for discoverable references using Web Services discovery.

**System.Web.Services.Discovery.ContractReference** ,  
**System.Web.Services.Discovery.SchemaReference** , and  
**System.Web.Services.Discovery.DiscoveryDocumentReference** all inherit from  
**System.Web.Services.Discovery.DiscoveryReference** , and represent the three  
types of documents discoverable through Web Services discovery: Service  
Descriptions, XML Schema Definition (XSD) schemas, and discovery documents,  
respectively.

DiscoveryReference

*Example Syntax:*

ToString

[C#] protected DiscoveryReference();

[C++] protected: DiscoveryReference();

[VB] Protected Sub New()

[JScript] protected function DiscoveryReference();

ClientProtocol

ToString

[C#] public DiscoveryClientProtocol ClientProtocol {get; set;}

[C++] public: \_\_property DiscoveryClientProtocol\* get\_ClientProtocol();public:

```

1 __property void set_ClientProtocol(DiscoveryClientProtocol*);
2 [VB] Public Property ClientProtocol As DiscoveryClientProtocol
3 [JScript] public function get ClientProtocol() : DiscoveryClientProtocol;public
4 function set ClientProtocol(DiscoveryClientProtocol);
5

```

#### *Description*

Gets or sets the instance of **System.Web.Services.Discovery.DiscoveryClientProtocol** used in a discovery process.

DefaultFilename

ToString

```

13 [C#] public virtual string DefaultFilename {get;}
14 [C++] public: __property virtual String* get_DefaultFilename();
15 [VB] Overridable Public ReadOnly Property DefaultFilename As String
16 [JScript] public function get DefaultFilename() : String;
17

```

#### *Description*

Gets the name of the default file to use when saving the referenced discovery document, XSD schema, or Service Description.

Url

ToString

```

24 [C#] public abstract string Url {get; set;}
25 [C++] public: __property virtual String* get_Url() = 0;public: __property virtual

```

1 void set\_Url(String\*) = 0;

2 [VB] MustOverride Public Property Url As String

3 [JScript] public abstract function get Url() : String; public abstract function set  
4 Url(String);

5  
6 *Description*

7 Gets or sets the URL of the referenced document.

8 FilenameFromUrl

9  
10 [C#] protected static string FilenameFromUrl(string url);

11 [C++] protected: static String\* FilenameFromUrl(String\* url);

12 [VB] Protected Shared Function FilenameFromUrl(ByVal url As String) As String

13 [JScript] protected static function FilenameFromUrl(url : String) : String;

14  
15 *Description*

16 Returns a file name based on the passed URL.

17 *Return Value:* Name of the file based on the passed URL. The URL on which the  
18 name of the file is based.

19 ReadDocument

20  
21 [C#] public abstract object ReadDocument(Stream stream);

22 [C++] public: virtual Object\* ReadDocument(Stream\* stream) = 0;

23 [VB] MustOverride Public Function ReadDocument(ByVal stream As Stream) As  
24 Object

25 [JScript] public abstract function ReadDocument(stream : Stream) : Object;

1  
2 *Description*

3 Reads the passed **System.IO.Stream** and returns an instance of the class  
4 representing the type of referenced document.

5 *Return Value:* An **System.Object** with an underlying type matching the type of  
6 referenced document.

7 The return value is always an **System.Object** . However, the underlying  
8 type varies, depending on the type of document referenced. For instance, a  
9 **System.IO.Stream** passed into  
10 **System.Web.Services.Discovery.SchemaReference.ReadDocument(System.IO**  
11 **.Stream)** returns an instance of **System.Xml.Schema.XmlSchema** .  
12 **System.IO.Stream** containing the reference document.

13 **Resolve**

14  
15 [C#] public void Resolve();

16 [C++] public: void Resolve();

17 [VB] Public Sub Resolve()

18 [JScript] public function Resolve(); Resolves whether the referenced document is  
19 valid.

20  
21 *Description*

22 Downloads the referenced document at  
23 **System.Web.Services.Discovery.DiscoveryReference.Url** to resolve whether the  
24 referenced document is valid.  
25



1 Downloads the referenced document and then invokes the overloaded  
2 **System.Web.Services.Discovery.DiscoveryReference.Resolve** member that  
3 takes a **contentType** and a **System.IO.Stream** . If the MIME type and the contents  
4 of *stream* match those expected by the reference type, then the contents of *stream*  
5 are added to the  
6 **System.Web.Services.Discovery.DiscoveryClientProtocol.References** and  
7 **System.Web.Services.Discovery.DiscoveryClientProtocol.Documents**  
8 properties of  
9 **System.Web.Services.Discovery.DiscoveryReference.ClientProtocol** .

#### 10 Resolve

11  
12 [C#] protected internal abstract void Resolve(string contentType, Stream stream);

13 [C++] protected public: virtual void Resolve(String\* contentType, Stream\*  
14 stream) = 0;

15 [VB] MustOverride Protected Friend Dim Sub Resolve(ByVal contentType As  
16 String, ByVal stream As Stream)

17 [JScript] package abstract function Resolve(contentType : String, stream :  
18 Stream);

#### 19 20 *Description*

21 Resolves whether the referenced document is valid.

22 If the MIME type and the contents of *stream* match those expected by the  
23 reference type, then the contents of *stream* are added to the  
24 **System.Web.Services.Discovery.DiscoveryClientProtocol.References** and  
25 **System.Web.Services.Discovery.DiscoveryClientProtocol.Documents**

properties of

**System.Web.Services.Discovery.DiscoveryReference.ClientProtocol** . The MIME type of *stream*. The **System.IO.Stream** containing the referenced document.

WriteDocument

[C#] public abstract void WriteDocument(object document, Stream stream);

[C++] public: virtual void WriteDocument(Object\* document, Stream\* stream) = 0;

[VB] MustOverride Public Sub WriteDocument(ByVal document As Object, ByVal stream As Stream)

[JScript] public abstract function WriteDocument(document : Object, stream : Stream);

### *Description*

When overridden in a derived class, writes the document into a **System.IO.Stream** . The document to write into a **System.IO.Stream**. The **System.IO.Stream** into which the *document* is written.

DiscoveryReferenceCollection class (System.Web.Services.Discovery)

WriteDocument

### *Description*

A collection of discovery references. This class cannot be inherited.

DiscoveryReferenceCollection

*Example Syntax:*

WriteDocument

[C#] public DiscoveryReferenceCollection();

[C++] public: DiscoveryReferenceCollection();

[VB] Public Sub New()

[JScript] public function DiscoveryReferenceCollection();

Count

InnerList

Item

WriteDocument

#### *Description*

Gets or sets the **System.Web.Services.Discovery.DiscoveryReference** at the specified index. The zero-based index of the **System.Web.Services.Discovery.DiscoveryReference** to get or set.

List

Add

[C#] public int Add(DiscoveryReference value);

[C++] public: int Add(DiscoveryReference\* value);

[VB] Public Function Add(ByVal value As DiscoveryReference) As Integer

[JScript] public function Add(value : DiscoveryReference) : int;

1  
2 *Description*

3 Adds a **System.Web.Services.Discovery.DiscoveryReference** to the  
4 **System.Web.Services.Discovery.DiscoveryReferenceCollection** .

5 *Return Value:* The position into which the  
6 **System.Web.Services.Discovery.DiscoveryReference** was inserted into the  
7 **System.Web.Services.Discovery.DiscoveryReferenceCollection** . The  
8 **System.Web.Services.Discovery.DiscoveryReference** to add to the  
9 **System.Web.Services.Discovery.DiscoveryReferenceCollection**.

10 Contains

11  
12 [C#] public bool Contains(DiscoveryReference value);  
13 [C++] public: bool Contains(DiscoveryReference\* value);  
14 [VB] Public Function Contains(ByVal value As DiscoveryReference) As Boolean  
15 [JScript] public function Contains(value : DiscoveryReference) : Boolean;  
16

17 *Description*

18 Determines whether the  
19 **System.Web.Services.Discovery.DiscoveryReferenceCollection** contains a  
20 specific **System.Web.Services.Discovery.DiscoveryReference** .

21 *Return Value:* **true** if the  
22 **System.Web.Services.Discovery.DiscoveryReferenceCollection** contains the  
23 **System.Web.Services.Discovery.DiscoveryReference** ; otherwise, **false** . The  
24 **System.Web.Services.Discovery.DiscoveryReference** to locate within the  
25 **System.Web.Services.Discovery.DiscoveryReferenceCollection**.

Remove

[C#] public void Remove(DiscoveryReference value);  
[C++] public: void Remove(DiscoveryReference\* value);  
[VB] Public Sub Remove(ByVal value As DiscoveryReference)  
[JScript] public function Remove(value : DiscoveryReference);

*Description*

Removes a **System.Web.Services.Discovery.DiscoveryReference** from the **System.Web.Services.Discovery.DiscoveryReferenceCollection**. The **System.Web.Services.Discovery.DiscoveryReference** to remove from the **System.Web.Services.Discovery.DiscoveryReferenceCollection**.

DiscoveryRequestHandler class (System.Web.Services.Discovery)

ToString

*Description*

DiscoveryRequestHandler

*Example Syntax:*

ToString

[C#] public DiscoveryRequestHandler();  
[C++] public: DiscoveryRequestHandler();

1 [VB] Public Sub New()  
2 [JScript] public function DiscoveryRequestHandler();  
3     IsReusable  
4     ToString  
5  
6 [C#] public bool IsReusable {get;}  
7 [C++] public: \_\_property bool get\_IsReusable();  
8 [VB] Public ReadOnly Property IsReusable As Boolean  
9 [JScript] public function get IsReusable() : Boolean;

10  
11 *Description*

12  
13     ProcessRequest  
14  
15 [C#] public void ProcessRequest(HttpContext context);  
16 [C++] public: \_\_sealed void ProcessRequest(HttpContext\* context);  
17 [VB] NotOverridable Public Sub ProcessRequest(ByVal context As HttpContext)  
18 [JScript] public function ProcessRequest(context : HttpContext);  
19

20 *Description*

21  
22     DiscoverySearchPattern class (System.Web.Services.Discovery)  
23     ToString  
24  
25

1  
2  
3 *Description*

4  
5       DiscoverySearchPattern

6       *Example Syntax:*

7       ToString

8  
9       [C#] protected DiscoverySearchPattern();

10       [C++] protected: DiscoverySearchPattern();

11       [VB] Protected Sub New()

12       [JScript] protected function DiscoverySearchPattern();

13       Pattern

14       ToString

15  
16       [C#] public abstract string Pattern {get;}

17       [C++] public: \_\_property virtual String\* get\_Pattern() = 0;

18       [VB] MustOverride Public ReadOnly Property Pattern As String

19       [JScript] public abstract function get Pattern() : String;

20  
21 *Description*

22  
23       GetDiscoveryReference

24  
25       [C#] public abstract DiscoveryReference GetDiscoveryReference(string filename);

```

1 [C++] public: virtual DiscoveryReference* GetDiscoveryReference(String*
2 filename) = 0;
3 [VB] MustOverride Public Function GetDiscoveryReference(ByVal filename As
4 String) As DiscoveryReference
5 [JScript] public abstract function GetDiscoveryReference(filename : String) :
6 DiscoveryReference;
7

```

#### *Description*

DynamicDiscoveryDocument class (System.Web.Services.Discovery)

ToString

#### *Description*

This represents a discovery file.

ToString

```

18 [C#] public const string Namespace;
19 [C++] public: const String* Namespace;
20 [VB] Public Const Namespace As String
21 [JScript] public var Namespace : String;
22

```

#### *Description*

DynamicDiscoveryDocument



*Example Syntax:*

ToString

```
[C#] public DynamicDiscoveryDocument();  
[C++] public: DynamicDiscoveryDocument();  
[VB] Public Sub New()  
[JScript] public function DynamicDiscoveryDocument();
```

*Description*

Default constructor.

ExcludePaths

ToString

```
[C#] public ExcludePathInfo[] ExcludePaths {get; set;}  
[C++] public: __property ExcludePathInfo* get_ExcludePaths();public:  
__property void set_ExcludePaths(ExcludePathInfo*[]);  
[VB] Public Property ExcludePaths As ExcludePathInfo ()  
[JScript] public function get ExcludePaths() : ExcludePathInfo[];public function  
set ExcludePaths(ExcludePathInfo[]);
```

*Description*

Load

```
[C#] public static DynamicDiscoveryDocument Load(Stream stream);
```

1 [C++] public: static DynamicDiscoveryDocument\* Load(Stream\* stream);  
 2 [VB] Public Shared Function Load(ByVal stream As Stream) As  
 3 DynamicDiscoveryDocument  
 4 [JScript] public static function Load(stream : Stream) :  
 5 DynamicDiscoveryDocument;

6  
 7 *Description*

8 Read an instance of WebMethodsFile from a stream.

9 Write

10  
 11 [C#] public void Write(Stream stream);  
 12 [C++] public: void Write(Stream\* stream);  
 13 [VB] Public Sub Write(ByVal stream As Stream)  
 14 [JScript] public function Write(stream : Stream);

15  
 16 *Description*

17 Write this instance to a stream.

18 ExcludePathInfo class (System.Web.Services.Discovery)

19 Write

20  
 21  
 22 *Description*

23  
 24 ExcludePathInfo

25 *Example Syntax:*

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

Write

```
[C#] public ExcludePathInfo();  
[C++] public: ExcludePathInfo();  
[VB] Public Sub New()  
[JScript] public function ExcludePathInfo();
```

*Description*

ExcludePathInfo

*Example Syntax:*

Write

```
[C#] public ExcludePathInfo(string path);  
[C++] public: ExcludePathInfo(String* path);  
[VB] Public Sub New(ByVal path As String)  
[JScript] public function ExcludePathInfo(path : String);
```

*Description*

Path

Write

```
[C#] public string Path {get; set;}  
[C++] public: __property String* get_Path();public: __property void
```

1 set\_Path(String\*);  
2 [VB] Public Property Path As String  
3 [JScript] public function get Path() : String;public function set Path(String);  
4

5 *Description*

6  
7 SchemaReference class (System.Web.Services.Discovery)  
8 ToString  
9

10  
11 *Description*

12 Represents a reference in a discovery document to an XML Schema  
13 Definition language (XSD) schema. This class cannot be inherited.

14 Web Services discovery involves discovering the available Web Services  
15 given an URL. A The URL typically points to a discovery document, that usually  
16 has a .disco file extension. The discovery document contains references to  
17 information about the existence of Web Services. These references can refer to  
18 Service Descriptions, XSD schemas, or other discovery documents. This class  
19 represents a reference to an XSD schema.

20 ToString  
21

22 [C#] public const string Namespace;  
23 [C++] public: const String\* Namespace;  
24 [VB] Public Const Namespace As String  
25 [JScript] public var Namespace : String;

1  
2 *Description*

3 XML namespace for XSD schema references in discovery documents.

4 Within a discovery document, a reference to an XSD schema is contained  
5 within a **schemaRef** XML element, which is a part of the XML namespace  
6 specified in the **System.Web.Services.Discovery.SchemaReference.Namespace**  
7 constant.

8 SchemaReference

9 *Example Syntax:*

10 ToString

11  
12 [C#] public SchemaReference();

13 [C++] public: SchemaReference();

14 [VB] Public Sub New()

15 [JScript] public function SchemaReference(); Initializes a new instance of the  
16 **System.Web.Services.Discovery.SchemaReference** class.

17  
18 *Description*

19 Initializes a new instance of the  
20 **System.Web.Services.Discovery.SchemaReference** class using default values.

21 SchemaReference

22 *Example Syntax:*

23 ToString

24  
25 [C#] public SchemaReference(string url);

1 [C++] public: SchemaReference(String\* url);  
2 [VB] Public Sub New(ByVal url As String)  
3 [JScript] public function SchemaReference(url : String);  
4

5 *Description*

6       Initializes a new instance of the  
7 **System.Web.Services.Discovery.SchemaReference** class using the supplied  
8 URL as the XSD schema reference. The URL for the XSD schema. Initializes the  
9 **System.Web.Services.Discovery.SchemaReference.Ref** property.

10       ClientProtocol

11       DefaultFilename

12       ToString  
13  
14

15 *Description*

16       Gets the name of the default file to use when saving the referenced XSD  
17 schema.

18       Ref

19       ToString  
20

21 [C#] public string Ref {get; set;}

22 [C++] public: \_\_property String\* get\_Ref();public: \_\_property void  
23 set\_Ref(String\*);

24 [VB] Public Property Ref As String

25 [JScript] public function get Ref() : String;public function set Ref(String);

## Description

Gets or sets the URL to the referenced XSD schema.

In a discovery document, a reference to an XSD schema is contained within a **schemaRef** XML element. The **schemaRef** XML element has a **ref** attribute, which is the URL for the referenced XSD schema. The **System.Web.Services.Discovery.ContractReference.Ref** property represents the value of the **ref** attribute.

Schema

ToString

[C#] public XmlSchema Schema {get;}

[C++] public: \_\_property XmlSchema\* get\_Schema();

[VB] Public ReadOnly Property Schema As XmlSchema

[JScript] public function get Schema() : XmlSchema;

## Description

Gets an **System.Xml.Schema.XmlSchema** object representing the XSD schema.

If the XSD schema has not been downloaded and added to the **System.Web.Services.Discovery.DiscoveryClientProtocol.Documents** property of **System.Web.Services.Discovery.DiscoveryReference.ClientProtocol**, an attempt to download and resolve the document is made.

TargetNamespace

ToString

```

1
2 [C#] public string TargetNamespace {get; set;}
3 [C++] public: __property String* get_TargetNamespace();public: __property void
4 set_TargetNamespace(String*);
5 [VB] Public Property TargetNamespace As String
6 [JScript] public function get TargetNamespace() : String;public function set
7 TargetNamespace(String);
8

```

### 9 *Description*

10 Gets or sets the **targetNamespace** XML attribute of the XSD schema.

11 Url

12 ToString

```

13
14 [C#] public override string Url {get; set;}
15 [C++] public: __property virtual String* get_Url();public: __property virtual void
16 set_Url(String*);
17 [VB] Overrides Public Property Url As String
18 [JScript] public function get Url() : String;public function set Url(String);
19

```

### 20 *Description*

21 Gets or sets the URL for the schema reference.

22 For the **System.Web.Services.Discovery.SchemaReference** class, the  
 23 **System.Web.Services.Discovery.SchemaReference.Url** property returns the  
 24 value of the **System.Web.Services.Discovery.SchemaReference.Ref** property.

25 ReadDocument



1 [C#] public override object ReadDocument(Stream stream);

2 [C++] public: Object\* ReadDocument(Stream\* stream);

3 [VB] Overrides Public Function ReadDocument(ByVal stream As Stream) As  
4 Object

5 [JScript] public override function ReadDocument(stream : Stream) : Object;

6  
7  
8 *Description*

9 Reads and returns the XSD schema from the passed **System.IO.Stream** .

10 *Return Value:* An **System.Xml.Schema.XmlSchema** containing the contents of  
11 the referenced XSD schema. **System.IO.Stream** containing the XSD schema.

12 **Resolve**

13  
14 [C#] protected internal override void Resolve(string contentType, Stream stream);

15 [C++] protected public: void Resolve(String\* contentType, Stream\* stream);

16 [VB] Overrides Protected Friend Dim Sub Resolve(ByVal contentType As String,  
17 ByVal stream As Stream)

18 [JScript] package override function Resolve(contentType : String, stream :  
19 Stream);

20  
21 *Description*

22 Resolves whether the the referenced document is valid.

23 If the MIME type is text/xml and the contents of *stream* are an XSD  
24 schema, then the contents of *stream* are added to the

25 **System.Web.Services.Discovery.DiscoveryClientProtocol.References** and

## **System.Web.Services.Discovery.DiscoveryClientProtocol.Documents**

properties of

**System.Web.Services.Discovery.DiscoveryReference.ClientProtocol** . The MIME content type of *stream*. The **System.IO.Stream** containing the referenced document.

### **WriteDocument**

[C#] public override void WriteDocument(object document, Stream stream);

[C++] public: void WriteDocument(Object\* document, Stream\* stream);

[VB] Overrides Public Sub WriteDocument(ByVal document As Object, ByVal stream As Stream)

[JScript] public override function WriteDocument(document : Object, stream : Stream);

### *Description*

Writes the passed XSD schema into the passed **System.IO.Stream** . The **System.Xml.Schema.XmlSchema** to write into *stream*. The **System.IO.Stream** into which the serialized XSD schema is written.

SoapBinding class (System.Web.Services.Discovery)

### **WriteDocument**

### *Description*

Represents a SOAP binding in a discovery document. This class cannot be inherited.

A SOAP binding is similar to an interface for Web Services and useful for versioning and when a developer is looking for a Web Service implementing a specific SOAP binding. For instance, a discovery document may contain optional information about SOAP Bindings, which specify the SOAP bindings implemented by referenced Web Services. SOAP bindings are specified within a discovery document by adding a **soap** XML element with an XML namespace equal to the **System.Web.Services.Discovery.SoapBinding.Namespace** constant. The **System.Web.Services.Discovery.SoapBinding.Address** property specifies the URL to the Web Service and the **System.Web.Services.Discovery.SoapBinding.Binding** property specifies the SOAP binding implemented by that Web Service.

WriteDocument

[C#] public const string Namespace;

[C++] public: const String\* Namespace;

[VB] Public Const Namespace As String

[JScript] public var Namespace : String;

### *Description*

The XML namespace of the element that specifies a SOAP binding within a discovery document.

SOAP bindings within a discovery document reside within a soap XML element that is a member of the XML namespace specified by the **System.Web.Services.Discovery.SoapBinding.Namespace** constant.

SoapBinding

*Example Syntax:*

WriteDocument

[C#] public SoapBinding();

[C++] public: SoapBinding();

[VB] Public Sub New()

[JScript] public function SoapBinding();

Address

WriteDocument

[C#] public string Address {get; set;}

[C++] public: \_\_property String\* get\_Address();public: \_\_property void

set\_Address(String\*);

[VB] Public Property Address As String

[JScript] public function get Address() : String;public function set

Address(String);

*Description*

Gets or sets the URL of the Web Service implementing the SOAP binding.

Binding

WriteDocument

[C#] public XmlQualifiedName Binding {get; set;}

[C++] public: \_\_property XmlQualifiedName\* get\_Binding();public: \_\_property

void set\_Binding(XmlQualifiedName\*);

1 [VB] Public Property Binding As XmlQualifiedName

2 [JScript] public function get Binding() : XmlQualifiedName;public function set

3 Binding(XmlQualifiedName);

4  
5 *Description*

6 Gets or sets the XML qualified name of the SOAP binding implemented by  
7 the Web Service.

8 XmlSchemaSearchPattern class (System.Web.Services.Discovery)

9 ToString

10  
11  
12 *Description*

13  
14 XmlSchemaSearchPattern

15 *Example Syntax:*

16 ToString

17  
18 [C#] public XmlSchemaSearchPattern();

19 [C++] public: XmlSchemaSearchPattern();

20 [VB] Public Sub New()

21 [JScript] public function XmlSchemaSearchPattern();

22 Pattern

23 ToString

24  
25 [C#] public override string Pattern {get;}

1 [C++] public: \_\_property virtual String\* get\_Pattern();

2 [VB] Overrides Public ReadOnly Property Pattern As String

3 [JScript] public function get Pattern() : String;

4  
5 *Description*

6  
7  
8  
9 **System.Web.Services.Protocols**

10  
11 *Description*

12 The **System.Web.Services.Protocols** namespace consists of the classes  
13 that define the protocols used to transmit data across the wire during the  
14 communication between ASP.NET Web Service clients and Web Services.

15 AnyReturnReader class (System.Web.Services.Protocols)

16  
17  
18 *Description*

19  
20 Constructors:

21 AnyReturnReader

22 *Example Syntax:*

23  
24 [C#] public AnyReturnReader();

25 [C++] public: AnyReturnReader();

1 [VB] Public Sub New()

2 [JScript] public function AnyReturnReader();

3       Methods:

4       GetInitializer

6 [C#] public override object GetInitializer(LogicalMethodInfo methodInfo);

7 [C++] public: Object\* GetInitializer(LogicalMethodInfo\* methodInfo);

8 [VB] Overrides Public Function GetInitializer(ByVal methodInfo As  
9 LogicalMethodInfo) As Object

10 [JScript] public override function GetInitializer(methodInfo : LogicalMethodInfo)  
11 : Object;

13 *Description*

15       Initialize

17 [C#] public override void Initialize(object o);

18 [C++] public: void Initialize(Object\* o);

19 [VB] Overrides Public Sub Initialize(ByVal o As Object)

20 [JScript] public override function Initialize(o : Object);

22 *Description*

24       Read

```

1
2 [C#] public override object Read(WebResponse response, Stream
3 responseStream);
4 [C++] public: Object* Read(WebResponse* response, Stream* responseStream);
5 [VB] Overrides Public Function Read(ByVal response As WebResponse, ByVal
6 responseStream As Stream) As Object
7 [JScript] public override function Read(response : WebResponse, responseStream
8 : Stream) : Object;
9

```

#### *Description*

HtmlFormParameterReader class (System.Web.Services.Protocols)

ToString

#### *Description*

HtmlFormParameterReader

*Example Syntax:*

ToString

```

22 [C#] public HtmlFormParameterReader();
23 [C++] public: HtmlFormParameterReader();
24 [VB] Public Sub New()
25 [JScript] public function HtmlFormParameterReader();

```



Read

[C#] public override object[] Read(HttpRequest request);  
[C++] public: Object\* Read(HttpRequest\* request) \_\_gc[];  
[VB] Overrides Public Function Read(ByVal request As HttpRequest) As Object()  
[JScript] public override function Read(request : HttpRequest) : Object[];

*Description*

HtmlFormParameterWriter class (System.Web.Services.Protocols)

ToString

*Description*

HtmlFormParameterWriter

*Example Syntax:*

ToString

[C#] public HtmlFormParameterWriter();  
[C++] public: HtmlFormParameterWriter();  
[VB] Public Sub New()  
[JScript] public function HtmlFormParameterWriter();

Properties:

RequestEncoding

UsesWriteRequest

ToString

*Description*

InitializeRequest

[C#] public override void InitializeRequest(WebRequest request, object[] values);

[C++] public: void InitializeRequest(WebRequest\* request, Object\* values  
\_\_gc[]);

[VB] Overrides Public Sub InitializeRequest(ByVal request As WebRequest,  
ByVal values() As Object)

[JScript] public override function InitializeRequest(request : WebRequest, values :  
Object[]);

*Description*

WriteRequest

[C#] public override void WriteRequest(Stream requestStream, object[] values);

[C++] public: void WriteRequest(Stream\* requestStream, Object\* values \_\_gc[]);

[VB] Overrides Public Sub WriteRequest(ByVal requestStream As Stream, ByVal  
values() As Object)

[JScript] public override function WriteRequest(requestStream : Stream, values :

Object[]);

*Description*

HttpGetClientProtocol class (System.Web.Services.Protocols)

WriteRequest

*Description*

Specifies the class for ASP.NET Web Service client proxies that use the HTTP-GET protocol.

ASP.NET incorporates two distinct Web Services functionalities: Building ASP.NET Web Services and Building Web Services clients. If you are building a Web Service client using ASP.NET, then a proxy class deriving indirectly or directly from **System.Web.Services.Protocols.WebClientProtocol** needs to be created for the Web Service you want to call. When the Web Service client calls the Web Service using HTTP, derive the proxy class from **System.Web.Services.Protocols.HttpSimpleClientProtocol**, which in turn derives from **System.Web.Services.Protocols.WebClientProtocol**.

HttpGetClientProtocol

*Example Syntax:*

WriteRequest

[C#] public HttpGetClientProtocol();

[C++] public: HttpGetClientProtocol();

1 [VB] Public Sub New()

2 [JScript] public function HttpGetClientProtocol();

3

4 *Description*

5       Initializes a new instance of the

6 **System.Web.Services.Protocols.HttpGetClientProtocol** class.

7       AllowAutoRedirect

8       ClientCertificates

9       ConnectionGroupName

10       Container

11       CookieContainer

12       Credentials

13       DesignMode

14       Events

15       PreAuthenticate

16       Proxy

17       RequestEncoding

18       Site

19       Timeout

20       Url

21       UserAgent

22       GetWebRequest

23

24 [C#] protected override WebRequest GetWebRequest(Uri uri);

25 [C++] protected: WebRequest\* GetWebRequest(Uri\* uri);

1 [VB] Overrides Protected Function GetWebRequest(ByVal uri As Uri) As  
2 WebRequest

3 [JScript] protected override function GetWebRequest(uri : Uri) : WebRequest;

4  
5 *Description*

6 Creates a **System.Net.WebRequest** instance for the specified URI.

7 *Return Value:* The **System.Net.WebRequest** instance.

8 This method overrides the base version of  
9 **System.Web.Services.Protocols.WebClientProtocol.GetWebRequest(System.**  
10 **Uri)** to specify that the HTTP request to the Web Service is made using HTTP-  
11 GET. By overriding this method, additional customizations can be made to the  
12 **System.Net.WebRequest** object before the Web Service request is made. For  
13 example you could add a custom header to the request. The **System.Uri** to use  
14 when creating the **System.Net.WebRequest**.

15 HttpMethodAttribute class (System.Web.Services.Protocols)

16 ToString

17  
18  
19 *Description*

20 Applying this attribute to a Web Service client using HTTP-GET or HTTP-  
21 POST, sets the types that serialize the parameters sent to a Web Service method  
22 and read the response from the Web Service method. This class cannot be  
23 inherited.

24 If a Web Service client invokes a Web Service method using HTTP-GET,  
25 **System.Web.Services.Protocols.HttpMethodAttribute.ReturnFormatter** must

1 be set to **XmlReturnReader** and

2 **System.Web.Services.Protocols.HttpMethodAttribute.ParameterFormatter**

3 set to **UrlParameterWriter** . Web Service clients invoking a Web Service using

4 HTTP-POST must set

5 **System.Web.Services.Protocols.HttpMethodAttribute.ReturnFormatter** to

6 **XmlReturnReader** and

7 **System.Web.Services.Protocols.HttpMethodAttribute.ParameterFormatter** to

8 **HtmlFormParameterWriter** .

9 **HttpMethodAttribute**

10 *Example Syntax:*

11 **ToString**

12  
13 [C#] public HttpMethodAttribute();

14 [C++] public: HttpMethodAttribute();

15 [VB] Public Sub New()

16 [JScript] public function HttpMethodAttribute(); Initializes a new instance of the

17 **System.Web.Services.Protocols.HttpMethodAttribute** class.

18  
19 *Description*

20 Initializes a new instance of the

21 **System.Web.Services.Protocols.HttpMethodAttribute** class.

22 **HttpMethodAttribute**

23 *Example Syntax:*

24 **ToString**

```

1
2 [C#] public HttpMethodAttribute(Type returnFormatter, Type
3 parameterFormatter);
4 [C++] public: HttpMethodAttribute(Type* returnFormatter, Type*
5 parameterFormatter);
6 [VB] Public Sub New(ByVal returnFormatter As Type, ByVal
7 parameterFormatter As Type)
8 [JScript] public function HttpMethodAttribute(returnFormatter : Type,
9 parameterFormatter : Type);
10

```

#### *Description*

Initializes a new instance of the  
**System.Web.Services.Protocols.HttpMethodAttribute** . Initializes the  
**System.Web.Services.Protocols.HttpMethodAttribute.ReturnFormatter**  
 property to a **System.Type** that deserializes the response from a Web Service  
 method. Initializes the  
**System.Web.Services.Protocols.HttpMethodAttribute.ParameterFormatter**  
 property to a **System.Type** that serializes parameters sent from a Web Service  
 client to a Web Service method.

ParameterFormatter

ToString

```

23 [C#] public Type ParameterFormatter {get; set;}
24 [C++] public: __property Type* get_ParameterFormatter();public: __property
25 void set_ParameterFormatter(Type*);

```

1 [VB] Public Property ParameterFormatter As Type

2 [JScript] public function get ParameterFormatter() : Type;public function set

3 ParameterFormatter(Type);

4  
5 *Description*

6 Gets or sets a **System.Type** that serializes parameters sent from a Web  
7 Service client to the Web Service method.

8 If the Web Service client is invoking a Web Service method using HTTP-  
9 GET or HTTP-POST,

10 **System.Web.Services.Protocols.HttpMethodAttribute.ParameterFormatter**  
11 must be **XMLReturnReader** .

12 ReturnFormatter

13 ToString

14  
15 [C#] public Type ReturnFormatter {get; set;}

16 [C++] public: \_\_property Type\* get\_ReturnFormatter();public: \_\_property void  
17 set\_ReturnFormatter(Type\*);

18 [VB] Public Property ReturnFormatter As Type

19 [JScript] public function get ReturnFormatter() : Type;public function set

20 ReturnFormatter(Type);

21  
22 *Description*

23 Gets or sets a **System.Type** that deserializes the response from a Web  
24 Service method.



1 If the Web Service client is invoking a Web Service method using HTTP-  
2 GET, **System.Web.Services.Protocols.HttpMethodAttribute.ReturnFormatter**  
3 must be **UrlParameterWriter** , whereas a client using HTTP-POST must set  
4 **System.Web.Services.Protocols.HttpMethodAttribute.ReturnFormatter** to  
5 **HtmlFormParameterWriter** .

6 **TypeId**

7 **HttpPostClientProtocol** class (**System.Web.Services.Protocols**)

8 **ToString**

9  
10  
11 *Description*

12 Specifies the class for ASP.NET Web Service client proxies that use the  
13 HTTP-POST protocol.

14 ASP.NET incorporates two distinct Web Services functionalities: Building  
15 ASP.NET Web Services and Building Web Services clients. If you are building a  
16 Web Service client using ASP.NET, then a proxy class deriving indirectly or  
17 directly from **System.Web.Services.Protocols.WebClientProtocol** needs to be  
18 created for the Web Service you want to call. When the Web Service client is  
19 calling using HTTP, the proxy class should derive from  
20 **System.Web.Services.Protocols.HttpSimpleClientProtocol** , which derives  
21 from **System.Web.Services.Protocols.WebClientProtocol** .

22 **HttpPostClientProtocol**

23 *Example Syntax:*

24 **ToString**

```

1
2 [C#] public HttpPostClientProtocol();
3 [C++] public: HttpPostClientProtocol();
4 [VB] Public Sub New()
5 [JScript] public function HttpPostClientProtocol();
6

```

### *Description*

Initializes a new instance of the

**System.Web.Services.Protocols.HttpPostClientProtocol** class.

AllowAutoRedirect

ClientCertificates

ConnectionGroupName

Container

CookieContainer

Credentials

DesignMode

Events

PreAuthenticate

Proxy

RequestEncoding

Site

Timeout

Url

UserAgent

GetWebRequest

1 [C#] protected override WebRequest GetWebRequest(Uri uri);  
 2  
 3 [C++] protected: WebRequest\* GetWebRequest(Uri\* uri);  
 4  
 5 [VB] Overrides Protected Function GetWebRequest(ByVal uri As Uri) As  
 6 WebRequest  
 7  
 8 [JScript] protected override function GetWebRequest(uri : Uri) : WebRequest;

#### 9 *Description*

10 Creates a **System.Net.WebRequest** instance for the specified URI.

11 *Return Value:* The **System.Net.WebRequest** instance.

12 This method overrides the base version of  
 13 **System.Web.Services.Protocols.WebClientProtocol.GetWebRequest(System.**  
 14 **Uri)** to specify that the HTTP request to the Web Service is made using HTTP-  
 15 POST. By overriding this method, you can customize the  
 16 **System.Net.WebRequest** object before the Web Service request is made. For  
 17 example, you can add a custom header to the request. The **System.Uri** to use when  
 18 creating the **System.Net.WebRequest**.

19 HttpSimpleClientProtocol class (System.Web.Services.Protocols)

20 ToString

#### 21 *Description*

22 The base class for communicating with an Web Service using HTTP-GET  
 23 and HTTP-POST.  
 24  
 25

1 Specifies most of the implementation for communicating with an Web  
2 Service over HTTP.

3 HttpSimpleClientProtocol

4 *Example Syntax:*

5 ToString

6  
7 [C#] protected HttpSimpleClientProtocol();

8 [C++] protected: HttpSimpleClientProtocol();

9 [VB] Protected Sub New()

10 [JScript] protected function HttpSimpleClientProtocol();

11  
12 *Description*

13 Initializes a new instance of the

14 **System.Web.Services.Protocols.HttpSimpleClientProtocol** class.

15 AllowAutoRedirect

16 ClientCertificates

17 ConnectionGroupName

18 Container

19 CookieContainer

20 Credentials

21 DesignMode

22 Events

23 PreAuthenticate

24 Proxy

25 RequestEncoding

1 Site

2 Timeout

3 Url

4 UserAgent

5 BeginInvoke

6  
7 [C#] protected IAsyncResult BeginInvoke(string methodName, string requestUrl,  
8 object[] parameters, AsyncCallback callback, object asyncState);

9 [C++] protected: IAsyncResult\* BeginInvoke(String\* methodName, String\*  
10 requestUrl, Object\* parameters \_\_gc[], AsyncCallback\* callback, Object\*  
11 asyncState);

12 [VB] Protected Function BeginInvoke(ByVal methodName As String, ByVal  
13 requestUrl As String, ByVal parameters() As Object, ByVal callback As  
14 AsyncCallback, ByVal asyncState As Object) As IAsyncResult

15 [JScript] protected function BeginInvoke(methodName : String, requestUrl :  
16 String, parameters : Object[], callback : AsyncCallback, asyncState : Object) :  
17 IAsyncResult;

18  
19 *Description*

20 Starts an asynchronous invocation of a method of a HTTP Web service.

21 *Return Value:* An **System.IAsyncResult** which can be passed to  
22 **System.Web.Services.Protocols.HttpSimpleClientProtocol.EndInvoke(System**  
23 **.IAsyncResult)** to obtain the return values from the Web Service method.

24 The *methodName* parameter is used to find the types of the parameters and  
25 return values of the method that is invoking

**System.Web.Services.Protocols.HttpSimpleClientProtocol.BeginInvoke(System.String, System.String, System.Object[], System.AsyncCallback, System.Object)**. It is also used to find custom attributes which may have been added to the method. The name of the Web Service method. The url to use when creating the **System.Net.WebRequest**. An array of objects containing the parameters to pass to the Web Service method. The order of the values in the array correspond to the order of the parameters in the calling method of the derived class. The delegate to call when the asynchronous method call is complete. If *callback* is **null**, the delegate is not called. Additional information supplied by a client.

#### EndInvoke

[C#] protected object EndInvoke(IAsyncResult asyncResult);

[C++] protected: Object\* EndInvoke(IAsyncResult\* asyncResult);

[VB] Protected Function EndInvoke(ByVal asyncResult As IAsyncResult) As Object

[JScript] protected function EndInvoke(asyncResult : IAsyncResult) : Object;

#### Description

Completes asynchronous invocation of a Web Service method using HTTP.

**Return Value:** An array of objects containing the return value and any by reference or out parameters for the Web Service method. The **System.IAsyncResult** returned from

**System.Web.Services.Protocols.HttpSimpleClientProtocol.BeginInvoke(System.String, System.String, System.Object[], System.AsyncCallback, System.Object)**.

## Invoke

```
[C#] protected object Invoke(string methodName, string requestUrl, object[]
parameters);
[C++] protected: Object* Invoke(String* methodName, String* requestUrl,
Object* parameters __gc[]);
[VB] Protected Function Invoke(ByVal methodName As String, ByVal requestUrl
As String, ByVal parameters() As Object) As Object
[JScript] protected function Invoke(methodName : String, requestUrl : String,
parameters : Object[]) : Object;
```

### *Description*

Invokes a Web Service method using HTTP.

*Return Value:* An array of objects containing the return value and any by-reference or out parameters of the derived class method.

The *methodName* is used to find the types of the parameters and return values of the method that is invoking

**System.Web.Services.Protocols.HttpSimpleClientProtocol.Invoke(System.String, System.String, System.Object[])** . It is also used to find custom attributes which may have been added to the method. The name of the Web Service method in the derived class that is invoking

**System.Web.Services.Protocols.HttpSimpleClientProtocol.Invoke(System.String, System.String, System.Object[])**. The URL of the Web Service method the client is requesting. An array of objects containing the parameters to pass to the

remote Web service. The order of the values in the array correspond to the order of the parameters in the calling method of the derived class.

HttpWebClientProtocol class (System.Web.Services.Protocols)

ToString

### *Description*

When overridden in a derived class, provides support for client proxies invoking Web Services using HTTP.

ASP.NET incorporates two distinct functionalities of Web Services: building ASP.NET Web Services and building Web Service clients. If you build a Web Service client using ASP.NET, you must create a proxy class deriving indirectly or directly from

**System.Web.Services.Protocols.HttpWebClientProtocol** for the Web Service you want to call.

HttpWebClientProtocol

*Example Syntax:*

ToString

[C#] protected HttpWebClientProtocol();

[C++] protected: HttpWebClientProtocol();

[VB] Protected Sub New()

[JScript] protected function HttpWebClientProtocol();

### *Description*



1        Initializes a new instance of the  
2        **System.Web.Services.Protocols.HttpWebClientProtocol** class.

3        AllowAutoRedirect

4        ToString

6        [C#] public bool AllowAutoRedirect {get; set;}

7        [C++] public: \_\_property bool get\_AllowAutoRedirect();public: \_\_property void  
8        set\_AllowAutoRedirect(bool);

9        [VB] Public Property AllowAutoRedirect As Boolean

10        [JScript] public function get AllowAutoRedirect() : Boolean;public function set  
11        AllowAutoRedirect(Boolean);

13        *Description*

14        Gets or sets whether the client automatically follows server redirects.

15        If a client sends authentication information, such as a user name and  
16        password, you do not want to enable the server to redirect, because this can  
17        compromise security.

18        ClientCertificates

19        ToString

21        [C#] public X509CertificateCollection ClientCertificates {get;}

22        [C++] public: \_\_property X509CertificateCollection\* get\_ClientCertificates();

23        [VB] Public ReadOnly Property ClientCertificates As X509CertificateCollection

24        [JScript] public function get ClientCertificates() : X509CertificateCollection;

1  
2 *Description*

3 Gets the collection of client certificates.

4 Allows a client to pass one or more client certificates, also known as  
5 Authenticode X.509 v.3 certificates, when calling a Web Service method. If the  
6 Web Service method has been configured to use client certificates, a client  
7 certificate can be used as one mechanism for authenticating a client. For details on  
8 setting up client certificates, see the Internet Information Server (IIS)  
9 documentation.

10 ConnectionGroupName

11 Container

12 CookieContainer

13 ToString

14  
15  
16 *Description*

17 Gets or sets the collection of cookies.

18 If a Web Service method uses session state, then a cookie is passed back to  
19 the Web Service client that uniquely identifies the session for that Web Service  
20 client. In order for the Web Service client to receive that cookie, a new instance of  
21 **System.Net.CookieContainer** must be created and assigned to the  
22 **System.Web.Services.Protocols.HttpWebClientProtocol.CookieContainer**  
23 property before calling the Web Service method.

24 Credentials

25 DesignMode

Events

PreAuthenticate

Proxy

ToString

*Description*

Gets or sets proxy information for making a Web Service request through a firewall.

Use the **System.Web.Services.Protocols.HttpWebClientProtocol.Proxy** property if a client needs to use different proxy settings than those in the system settings. You can use the **System.Net.WebProxy** class to set the proxy settings, because it implements **System.Net.IWebProxy**.

RequestEncoding

Site

Timeout

Url

UserAgent

ToString

*Description*

Gets or sets the value for the user agent header that is sent with each request.

The user agent string allows a Web server to identify the client.

## GetWebRequest

[C#] protected override WebRequest GetWebRequest(Uri uri);

[C++] protected: WebRequest\* GetWebRequest(Uri\* uri);

[VB] Overrides Protected Function GetWebRequest(ByVal uri As Uri) As WebRequest

[JScript] protected override function GetWebRequest(uri : Uri) : WebRequest;

### *Description*

Creates a **System.Net.WebRequest** instance for the specified URI.

*Return Value:* The **System.Net.WebRequest** instance. The **System.Uri** for creating the **System.Net.WebRequest**.

## GetWebResponse

[C#] protected override WebResponse GetWebResponse(WebRequest request);

[C++] protected: WebResponse\* GetWebResponse(WebRequest\* request);

[VB] Overrides Protected Function GetWebResponse(ByVal request As WebRequest) As WebResponse

[JScript] protected override function GetWebResponse(request : WebRequest) : WebResponse; Returns a response from a request to a Web Service method.

### *Description*

Returns a response from a synchronous request to a Web Service method.

*Return Value:* The **System.Net.WebResponse** instance. The **System.Net.WebRequest** to get the response from.

## GetWebResponse

[C#] protected override WebResponse GetWebResponse(WebRequest request, IAsyncResult result);

[C++] protected: WebResponse\* GetWebResponse(WebRequest\* request, IAsyncResult\* result);

[VB] Overrides Protected Function GetWebResponse(ByVal request As WebRequest, ByVal result As IAsyncResult) As WebResponse

[JScript] protected override function GetWebResponse(request : WebRequest, result : IAsyncResult) : WebResponse;

### *Description*

Returns a response from an asynchronous request to a Web Service method.

**Return Value:** The **System.Net.WebResponse** instance. The **System.Net.WebRequest** to get the response from. The **System.IAsyncResult** to pass to **System.Net.HttpWebRequest.EndGetResponse(System.IAsyncResult)**. when the response has completed.

LogicalMethodInfo class (System.Web.Services.Protocols)

ToString

### *Description*

Represents the attributes and metadata for a Web Service method. This class cannot be inherited.

**System.Web.Services.Protocols.LogicalMethodInfo** is used primarily by a SOAP extension to interrogate the details of the Web Service method an SOAP extension is configured to run with. Depending on how the SOAP extension is configured to run, the SOAP extension can find out details about the Web Service method in the

**System.Web.Services.Protocols.SoapExtension.GetInitializer(System.Web.Services.Protocols.LogicalMethodInfo, System.Web.Services.Protocols.SoapExtensionAttribute)** method of **System.Web.Services.Protocols.SoapExtension** that takes an **System.Web.Services.Protocols.LogicalMethodInfo**. The **System.Web.Services.Protocols.LogicalMethodInfo** provides details such as the Web Service method's parameters by accessing the **System.Web.Services.Protocols.LogicalMethodInfo.Parameters** property and any custom attributes applied to the Web Service method using the **System.Web.Services.Protocols.LogicalMethodInfo.GetCustomAttributes(System.Type)** property.

**LogicalMethodInfo**

*Example Syntax:*

**ToString**

[C#] public LogicalMethodInfo(MethodInfo methodInfo);

[C++] public: LogicalMethodInfo(MethodInfo\* methodInfo);

[VB] Public Sub New(ByVal methodInfo As MethodInfo)

[JScript] public function LogicalMethodInfo(methodInfo : MethodInfo); Initializes a new instance of the **System.Web.Services.Protocols.LogicalMethodInfo** class.

1  
2 *Description*

3       Initializes a new instance of the  
4 **System.Web.Services.Protocols.LogicalMethodInfo** class with the  
5 **System.Reflection.MethodInfo** object passed. A **System.Reflection.MethodInfo**  
6 to initialize the properties of  
7 **System.Web.Services.Protocols.LogicalMethodInfo** common to  
8 **System.Reflection.MethodInfo**.

9       AsyncCallbackParameter

10      ToString

11  
12 [C#] public ParameterInfo AsyncCallbackParameter {get;}

13 [C++] public: \_\_property ParameterInfo\* get\_AsyncCallbackParameter();

14 [VB] Public ReadOnly Property AsyncCallbackParameter As ParameterInfo

15 [JScript] public function get AsyncCallbackParameter() : ParameterInfo;

16  
17 *Description*

18       Gets the parameter information for the *AsyncCallback* parameter of a *Begin*  
19 method in an asynchronous invocation.

20       The asynchronous design pattern in the common language runtime involves  
21 calling a **Begin** method to start the asynchronous method invocation and an **End**  
22 method to complete the invocation. The **Begin** method takes two additional  
23 parameters besides the parameters defined by the method: one for a delegate and  
24 one for any state information that needs to be passed on to the delegate. This  
25

property represents the parameter for the delegate with a parameter name of *AsyncCallback* .

*AsyncResultParameter*

*ToString*

[C#] public ParameterInfo AsyncResultParameter {get;}

[C++] public: \_\_property ParameterInfo\* get\_AsyncResultParameter();

[VB] Public ReadOnly Property AsyncResultParameter As ParameterInfo

[JScript] public function get AsyncResultParameter() : ParameterInfo;

### *Description*

Gets the return value of a **Begin** asynchronous method invocation.

The asynchronous design pattern in the common language runtime involves calling a **Begin** method to start the asynchronous method invocation and an **End** method to complete the invocation. The **Begin** method typically returns immediately with an object implementing the **System.IAsyncResult** interface, which can then be passed to the **End** method at a later time to complete the asynchronous method invocation. The returned object implementing the **System.IAsyncResult** interface is represented by this property.

*AsyncStateParameter*

*ToString*

[C#] public ParameterInfo AsyncStateParameter {get;}

[C++] public: \_\_property ParameterInfo\* get\_AsyncStateParameter();

[VB] Public ReadOnly Property AsyncStateParameter As ParameterInfo



1 [JScript] public function get AsyncStateParameter() : ParameterInfo;

3 *Description*

4 Gets the parameter information for the *AsyncState* parameter of a **Begin**  
5 method in an asynchronous invocation.

6 The asynchronous design pattern in the common language runtime involves  
7 calling a **Begin** method to start the asynchronous method invocation and an **End**  
8 method to complete the invocation. The **Begin** method takes two additional  
9 parameters besides the parameters defined by the method: one for a delegate and  
10 one for any state information that needs to be passed on to the delegate. This  
11 property represents the state information that needs to be passed into the delegate.

12 BeginMethodInfo

13 ToString

15 [C#] public MethodInfo BeginMethodInfo {get;}

16 [C++] public: \_\_property MethodInfo\* get \_BeginMethodInfo();

17 [VB] Public ReadOnly Property BeginMethodInfo As MethodInfo

18 [JScript] public function get BeginMethodInfo() : MethodInfo;

20 *Description*

21 Gets the attributes and metadata for a **Begin** method of an asynchronous  
22 invocation to a method.

23 CustomAttributeProvider

24 ToString

1 [C#] public ICustomAttributeProvider CustomAttributeProvider {get;}

2 [C++] public: \_\_property ICustomAttributeProvider\*

3 get\_CustomAttributeProvider();

4 [VB] Public ReadOnly Property CustomAttributeProvider As

5 ICustomAttributeProvider

6 [JScript] public function get CustomAttributeProvider() :

7 ICustomAttributeProvider;

8  
9  
10 *Description*

11 Gets the custom attributes applied to the method.

12 DeclaringType

13 ToString

14  
15 [C#] public Type DeclaringType {get;}

16 [C++] public: \_\_property Type\* get\_DeclaringType();

17 [VB] Public ReadOnly Property DeclaringType As Type

18 [JScript] public function get DeclaringType() : Type;

19  
20 *Description*

21 Gets the class that declares the method represented by the instance of

22 **System.Web.Services.Protocols.LogicalMethodInfo** .

23 The **System.Web.Services.Protocols.LogicalMethodInfo.DeclaringType**  
24 property retrieves a reference to a **System.Type** for the type that declares this  
25 member. A member of a class (or interface) is either declared or inherited from a

base class (or interface). The returned **System.Type** might not be the same as the **System.Type** of the class implementing the Web Service (if that class derives from a base class and the method represented by this class is declared in that base class then the **System.Type** returned is the base class).

EndMethodInfo

ToString

[C#] public MethodInfo EndMethodInfo {get;}

[C++] public: \_\_property MethodInfo\* get\_EndMethodInfo();

[VB] Public ReadOnly Property EndMethodInfo As MethodInfo

[JScript] public function get EndMethodInfo() : MethodInfo;

### *Description*

Gets the attributes and metadata for an **End** method of an asynchronous invocation to a method.

InParameters

ToString

[C#] public ParameterInfo[] InParameters {get;}

[C++] public: \_\_property ParameterInfo\* get\_InParameters();

[VB] Public ReadOnly Property InParameters As ParameterInfo ()

[JScript] public function get InParameters() : ParameterInfo[];

### *Description*

1 Gets the parameters passed into the method represented by the instance of  
2 **System.Web.Services.Protocols.LogicalMethodInfo** .

3 Use an instance of **System.Reflection.ParameterInfo** to obtain  
4 information about the parameter's data type, default value, and so on.

5 IsAsync

6 ToString

7  
8 [C#] public bool IsAsync {get;}

9 [C++] public: \_\_property bool get\_IsAsync();

10 [VB] Public ReadOnly Property IsAsync As Boolean

11 [JScript] public function get IsAsync() : Boolean;

12  
13 *Description*

14 Gets a value indicating whether the method represented by the instance of  
15 **System.Web.Services.Protocols.LogicalMethodInfo** is invoked asynchronously.

16 IsVoid

17 ToString

18  
19 [C#] public bool IsVoid {get;}

20 [C++] public: \_\_property bool get\_IsVoid();

21 [VB] Public ReadOnly Property IsVoid As Boolean

22 [JScript] public function get IsVoid() : Boolean;

23  
24 *Description*

1 Gets a value indicating whether the return type for the method represented  
2 by the instance of **System.Web.Services.Protocols.LogicalMethodInfo** is **void** .

3 **MethodInfo**

4 **ToString**

5  
6 [C#] public MethodInfo MethodInfo {get;}

7 [C++] public: \_\_property MethodInfo\* get\_MethodInfo();

8 [VB] Public ReadOnly Property MethodInfo As MethodInfo

9 [JScript] public function get MethodInfo() : MethodInfo;

10  
11 *Description*

12 Gets the attributes and metadata for a synchronous method.

13 **Name**

14 **ToString**

15  
16 [C#] public string Name {get;}

17 [C++] public: \_\_property String\* get\_Name();

18 [VB] Public ReadOnly Property Name As String

19 [JScript] public function get Name() : String;

20  
21 *Description*

22 Gets the name of the method represented by this instance.

23 **OutParameters**

24 **ToString**

```

1
2 [C#] public ParameterInfo[] OutParameters {get;}
3 [C++] public: __property ParameterInfo* get_OutParameters();
4 [VB] Public ReadOnly Property OutParameters As ParameterInfo ()
5 [JScript] public function get OutParameters() : ParameterInfo[];
6

```

### *Description*

Gets the out parameters for the method.

Use an instance of **System.Reflection.ParameterInfo** to obtain information about the parameter's data type, default value, and so on.

Parameters

ToString

```

11
12
13
14 [C#] public ParameterInfo[] Parameters {get;}
15 [C++] public: __property ParameterInfo* get_Parameters();
16 [VB] Public ReadOnly Property Parameters As ParameterInfo ()
17 [JScript] public function get Parameters() : ParameterInfo[];
18

```

### *Description*

Gets the parameters for the method.

Use an instance of **System.Reflection.ParameterInfo** to obtain information about the parameter's data type, default value, and so on.

ReturnType

ToString

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#] public Type ReturnType {get;}
[C++] public: __property Type* get_ReturnType();
[VB] Public ReadOnly Property ReturnType As Type
[JScript] public function get ReturnType() : Type;
```

*Description*

Gets the return type of this method.

ReturnTypeCustomAttributeProvider

ToString

```
[C#] public ICustomAttributeProvider ReturnTypeCustomAttributeProvider
{get;}
[C++] public: __property ICustomAttributeProvider*
get_ReturnTypeCustomAttributeProvider();
[VB] Public ReadOnly Property ReturnTypeCustomAttributeProvider As
ICustomAttributeProvider
[JScript] public function get ReturnTypeCustomAttributeProvider() :
ICustomAttributeProvider;
```

*Description*

Gets the custom attributes for the return type.

BeginInvoke

```
[C#] public IAsyncResult BeginInvoke(object target, object[] values,
```

```

1 AsyncCallback callback, object asyncState);
2 [C++] public: IAsyncResult* BeginInvoke(Object* target, Object* values __gc[],
3 AsyncCallback* callback, Object* asyncState);
4 [VB] Public Function BeginInvoke(ByVal target As Object, ByVal values() As
5 Object, ByVal callback As AsyncCallback, ByVal asyncState As Object) As
6 IAsyncResult
7 [JScript] public function BeginInvoke(target : Object, values : Object[], callback :
8 AsyncCallback, asyncState : Object) : IAsyncResult;
9

```

### *Description*

Begins an asynchronous invocation of the method represented by this **System.Web.Services.Protocols.LogicalMethodInfo** instance.

*Return Value:* An **System.IAsyncResult** which is passed to **System.Web.Services.Protocols.LogicalMethodInfo.EndInvoke(System.Object, System.IAsyncResult)** to obtain the return values from the remote method call. The instance of the object to invoke the method on. An argument list for the invoked method. This is an array of objects with the same number, order, and type as the parameters of the method. If the method does not require any parameters, *values* should be **null**. The delegate to call when the asynchronous invoke is complete. If *callback* is **null**, the delegate is not called. State information that is passed on to the delegate.

### *Create*

```

24 [C#] public static LogicalMethodInfo[] Create(MethodInfo[] methodInfos);
25 [C++] public: static LogicalMethodInfo* Create(MethodInfo* methodInfos[]) [];

```



1 [VB] Public Shared Function Create(ByVal methodInfos() As MethodInfo) As  
2 LogicalMethodInfo()

3 [JScript] public static function Create(methodInfos : MethodInfo[]) :  
4 LogicalMethodInfo[]; Creates an array of  
5 **System.Web.Services.Protocols.LogicalMethodInfo** , given an array of  
6 **System.Reflection.MethodInfo** .

7  
8 *Description*

9 Creates an array of **System.Web.Services.Protocols.LogicalMethodInfo** ,  
10 given an array of **System.Reflection.MethodInfo** that can contain information  
11 about both asynchronous and synchronous methods.

12 *Return Value:* An array of **System.Web.Services.Protocols.LogicalMethodInfo** ,  
13 representing the methods within *methodInfos* . An array of  
14 **System.Reflection.MethodInfo** representing the asynchronous and synchronous  
15 methods to create **System.Web.Services.Protocols.LogicalMethodInfo** objects  
16 for.

17 **Create**

18  
19 [C#] public static LogicalMethodInfo[] Create(MethodInfo[] methodInfos,  
20 LogicalMethodTypes types);

21 [C++] public: static LogicalMethodInfo\* Create(MethodInfo\* methodInfos[],  
22 LogicalMethodTypes types) [];

23 [VB] Public Shared Function Create(ByVal methodInfos() As MethodInfo, ByVal  
24 types As LogicalMethodTypes) As LogicalMethodInfo()

25 [JScript] public static function Create(methodInfos : MethodInfo[], types :

LogicalMethodTypes) : LogicalMethodInfo[];

### *Description*

Creates an array of **System.Web.Services.Protocols.LogicalMethodInfo** , given an array of **System.Reflection.MethodInfo** , where the returned array of **System.Web.Services.Protocols.LogicalMethodInfo** can be restricted to only asynchronous or synchronous methods.

*Return Value:* An array of **System.Web.Services.Protocols.LogicalMethodInfo** representing the methods within *methodInfos*, filtered by the value of *types* . An array of **System.Reflection.MethodInfo** representing the asynchronous and synchronous methods to create **System.Web.Services.Protocols.LogicalMethodInfo** objects for. A bitwise combination of the **System.Web.Services.Protocols.LogicalMethodTypes** values. Determines whether just asynchronous or synchronous methods or both are included in the returned array of **System.Web.Services.Protocols.LogicalMethodInfo**.

EndInvoke

[C#] public object[] EndInvoke(object target, IAsyncResult asyncResult);

[C++] public: Object\* EndInvoke(Object\* target, IAsyncResult\* asyncResult)

\_\_gc[];

[VB] Public Function EndInvoke(ByVal target As Object, ByVal asyncResult As IAsyncResult) As Object()

[JScript] public function EndInvoke(target : Object, asyncResult : IAsyncResult) : Object[];

## Description

Ends an asynchronous invocation of the method represented by this

**System.Web.Services.Protocols.LogicalMethodInfo** instance.

*Return Value:* An array of objects containing the return value and any by-reference or out parameters of the derived class method. The instance of the object to invoke the method on. The **System.IAsyncResult** returned from **System.Web.Services.Protocols.LogicalMethodInfo.BeginInvoke(System.Object, System.Object[], System.AsyncCallback, System.Object)**.

## GetCustomAttribute

[C#] public object GetCustomAttribute(Type type);

[C++] public: Object\* GetCustomAttribute(Type\* type);

[VB] Public Function GetCustomAttribute(ByVal type As Type) As Object

[JScript] public function GetCustomAttribute(type : Type) : Object;

## Description

Returns the first custom attribute applied to the type, if there are custom attributes applied to the type.

*Return Value:* An **System.Object** containing the first custom attribute applied to *type*. The **System.Type** that the custom attributes are applied to.

## GetCustomAttributes

[C#] public object[] GetCustomAttributes(Type type);

[C++] public: Object\* GetCustomAttributes(Type\* type) \_\_gc[];

1 [VB] Public Function GetCustomAttributes(ByVal type As Type) As Object()

2 [JScript] public function GetCustomAttributes(type : Type) : Object[];

3  
4 *Description*

5 Returns the custom attributes applied to the specified type.

6 *Return Value:* An array of **System.Object** containing the custom attributes applied  
7 to *type* . The **System.Type** to get the custom attributes applied to.

8 **Invoke**

9  
10 [C#] public object[] Invoke(object target, object[] values);

11 [C++] public: Object\* Invoke(Object\* target, Object\* values \_\_gc[]) \_\_gc[];

12 [VB] Public Function Invoke(ByVal target As Object, ByVal values() As Object)  
13 As Object()

14 [JScript] public function Invoke(target : Object, values : Object[]) : Object[];

15  
16 *Description*

17 Invokes the method represented by this

18 **System.Web.Services.Protocols.LogicalMethodInfo** instance.

19 *Return Value:* An array of type **System.Object** representing the return value and  
20 out parameters of the invoked method. The instance of the object to invoke the  
21 method on. An argument list for the invoked method. This is an array of objects  
22 with the same number, order, and type as the parameters of the method. If the  
23 method does not require any parameters, *values* should be **null**.

24 **IsBeginMethod**

```

1
2 [C#] public static bool IsBeginMethod(MethodInfo methodInfo);
3 [C++] public: static bool IsBeginMethod(MethodInfo* methodInfo);
4 [VB] Public Shared Function IsBeginMethod(ByVal methodInfo As MethodInfo)
5 As Boolean
6 [JScript] public static function IsBeginMethod(methodInfo : MethodInfo) :
7 Boolean;
8

```

### *Description*

Returns a value indicating whether the method passed in represents a begin method of an asynchronous invocation.

*Return Value:* **true** if *methodInfo* is a begin method of an asynchronous invocation; otherwise, **false** . The **System.Reflection.MethodInfo** to determine if it is a begin method of an asynchronous invocation.

### *IsEndMethod*

```

17 [C#] public static bool IsEndMethod(MethodInfo methodInfo);
18 [C++] public: static bool IsEndMethod(MethodInfo* methodInfo);
19 [VB] Public Shared Function IsEndMethod(ByVal methodInfo As MethodInfo)
20 As Boolean
21 [JScript] public static function IsEndMethod(methodInfo : MethodInfo) : Boolean;
22

```

### *Description*

Returns a value indicating whether the method passed in represents a end method of an asynchronous invocation.

*Return Value:* **true** if *MethodInfo* is a end method of an asynchronous invocation; otherwise, **false** . The **System.Reflection.MethodInfo** to determine if it is a end method of an asynchronous invocation.

ToString

[C#] public override string ToString();

[C++] public: String\* ToString();

[VB] Overrides Public Function ToString() As String

[JScript] public override function ToString() : String;

#### *Description*

Returns a **System.String** that represents the current **System.Web.Services.Protocols.LogicalMethodInfo** .

*Return Value:* A **System.String** that represents the current **System.Web.Services.Protocols.LogicalMethodInfo** .

LogicalMethodTypes enumeration (System.Web.Services.Protocols)

ToString

#### *Description*

Specifies how the Web Service method was invoked.

ToString

[C#] public const LogicalMethodTypes Async;

[C++] public: const LogicalMethodTypes Async;

1 [VB] Public Const Async As LogicalMethodTypes

2 [JScript] public var Async : LogicalMethodTypes;

3  
4 *Description*

5 Specifies the Web Service method is invoked asynchronously.

6 ToString

7  
8 [C#] public const LogicalMethodTypes Sync;

9 [C++] public: const LogicalMethodTypes Sync;

10 [VB] Public Const Sync As LogicalMethodTypes

11 [JScript] public var Sync : LogicalMethodTypes;

12  
13 *Description*

14 Specifies the Web Service method is invoked synchronously.

15 MatchAttribute class (System.Web.Services.Protocols)

16 ToString

17  
18  
19 *Description*

20 Represents the attributes of a match made using text pattern matching. This  
21 class cannot be inherited.

22 Text pattern matching allows a Web Service to leverage existing HTML  
23 content by parsing it using regular expressions. A Web Service specifies the  
24 content it wants to parse in a Service Description using match elements. These  
25 match elements specify several items: the regular expression for parsing the

contents of an existing HTML page, whether the parsing must case-insensitive, and how many instances of content that matches the regular expression should be returned. When a client builds a proxy class using the Wsdl.exe tool, methods on the proxy class include a **System.Web.Services.Protocols.MatchAttribute** detailing the match elements found in the Service Description.

MatchAttribute

*Example Syntax:*

ToString

[C#] public MatchAttribute(string pattern);

[C++] public: MatchAttribute(String\* pattern);

[VB] Public Sub New(ByVal pattern As String)

[JScript] public function MatchAttribute(pattern : String);

### Description

Initializes a new instance of the

**System.Web.Services.Protocols.MatchAttribute** class with the specified pattern.

Use this constructor to create and initialize a new instance of the

**System.Web.Services.Protocols.MatchAttribute** class using the specified pattern. A string that represents the pattern to match.

Capture

ToString

[C#] public int Capture {get; set;}

[C++] public: \_\_property int get\_Capture();public: \_\_property void



1 set\_Capture(int);

2 [VB] Public Property Capture As Integer

3 [JScript] public function get Capture() : int;public function set Capture(int);

4  
5 *Description*

6 Gets or sets a value that represents the index of a match within a grouping.

7 Group

8 ToString

9  
10 [C#] public int Group {get; set;}

11 [C++] public: \_\_property int get\_Group();public: \_\_property void set\_Group(int);

12 [VB] Public Property Group As Integer

13 [JScript] public function get Group() : int;public function set Group(int);

14  
15 *Description*

16 Gets or sets a value that represents a grouping of related matches.

17 Use the **System.Web.Services.Protocols.MatchAttribute.Group** property  
18 to specify a group that a match belongs to.

19 IgnoreCase

20 ToString

21  
22 [C#] public bool IgnoreCase {get; set;}

23 [C++] public: \_\_property bool get\_IgnoreCase();public: \_\_property void  
24 set\_IgnoreCase(bool);

25 [VB] Public Property IgnoreCase As Boolean

[JScript] public function get IgnoreCase() : Boolean;public function set IgnoreCase(Boolean);

#### *Description*

Gets or sets a value that indicates whether the pattern to match is case insensitive.

MaxRepeats

ToString

[C#] public int MaxRepeats {get; set;}

[C++] public: \_\_property int get\_MaxRepeats();public: \_\_property void set\_MaxRepeats(int);

[VB] Public Property MaxRepeats As Integer

[JScript] public function get MaxRepeats() : int;public function set MaxRepeats(int);

#### *Description*

Gets or sets the maximum number of values to return from the match.

A value of 1 returns only the first match. A value of -1 returns all matches.

Additionally, a value of -1 equates to an '\*' in a regular expression.

Pattern

ToString

[C#] public string Pattern {get; set;}

[C++] public: \_\_property String\* get\_Pattern();public: \_\_property void

```

1 set_Pattern(String*);
2 [VB] Public Property Pattern As String
3 [JScript] public function get Pattern() : String;public function set Pattern(String);
4

```

#### *Description*

Gets or sets a regular expression that represents the pattern to match.

Use the **System.Web.Services.Protocols.MatchAttribute.Pattern** property to specify a regular expression that represents the value to match.

TypeId

MimeFormatter class (System.Web.Services.Protocols)

ToString

#### *Description*

MimeFormatter

*Example Syntax:*

ToString

```

20 [C#] protected MimeFormatter();
21 [C++] protected: MimeFormatter();
22 [VB] Protected Sub New()
23 [JScript] protected function MimeFormatter();
24
25 CreateInstance

```

1  
2 [C#] public static MimeFormatter CreateInstance(Type type, object initializer);  
3 [C++] public: static MimeFormatter\* CreateInstance(Type\* type, Object\*  
4 initializer);  
5 [VB] Public Shared Function CreateInstance(ByVal type As Type, ByVal  
6 initializer As Object) As MimeFormatter  
7 [JScript] public static function CreateInstance(type : Type, initializer : Object) :  
8 MimeFormatter;

9  
10 *Description*

11  
12 **GetInitializer**

13  
14 [C#] public abstract object GetInitializer(LogicalMethodInfo methodInfo);  
15 [C++] public: virtual Object\* GetInitializer(LogicalMethodInfo\* methodInfo) = 0;  
16 [VB] MustOverride Public Function GetInitializer(ByVal methodInfo As  
17 LogicalMethodInfo) As Object  
18 [JScript] public abstract function GetInitializer(methodInfo : LogicalMethodInfo) :  
19 Object;

20  
21 *Description*

22  
23 **GetInitializer**

24  
25 [C#] public static object GetInitializer(Type type, LogicalMethodInfo

1 methodInfo);

2 [C++] public: static Object\* GetInitializer(Type\* type, LogicalMethodInfo\*

3 methodInfo);

4 [VB] Public Shared Function GetInitializer(ByVal type As Type, ByVal

5 methodInfo As LogicalMethodInfo) As Object

6 [JScript] public static function GetInitializer(type : Type, methodInfo :

7 LogicalMethodInfo) : Object;

8  
9 *Description*

## 10 11 GetInitializers

12  
13 [C#] public virtual object[] GetInitializers(LogicalMethodInfo[] methodInfos);

14 [C++] public: virtual Object\* GetInitializers(LogicalMethodInfo\* methodInfos[])

15 \_\_gc[];

16 [VB] Overridable Public Function GetInitializers(ByVal methodInfos() As

17 LogicalMethodInfo) As Object()

18 [JScript] public function GetInitializers(methodInfos : LogicalMethodInfo[]) :

19 Object[];

20  
21 *Description*

## 22 23 GetInitializers

24  
25 [C#] public static object[] GetInitializers(Type type, LogicalMethodInfo[]

```

1 methodInfos);
2 [C++] public: static Object* GetInitializers(Type* type, LogicalMethodInfo*
3 methodInfos[]) __gc[];
4 [VB] Public Shared Function GetInitializers(ByVal type As Type, ByVal
5 methodInfos() As LogicalMethodInfo) As Object()
6 [JScript] public static function GetInitializers(type : Type, methodInfos :
7 LogicalMethodInfo[]) : Object[];
8

```

### *Description*

#### Initialize

```

13 [C#] public abstract void Initialize(object initializer);
14 [C++] public: virtual void Initialize(Object* initializer) = 0;
15 [VB] MustOverride Public Sub Initialize(ByVal initializer As Object)
16 [JScript] public abstract function Initialize(initializer : Object);
17

```

### *Description*

MimeParameterReader class (System.Web.Services.Protocols)

#### ToString

### *Description*

MimeParameterReader

*Example Syntax:*

ToString

[C#] protected MimeParameterReader();

[C++] protected: MimeParameterReader();

[VB] Protected Sub New()

[JScript] protected function MimeParameterReader();

Read

[C#] public abstract object[] Read(HttpRequest request);

[C++] public: virtual Object\* Read(HttpRequest\* request) \_\_gc[] = 0;

[VB] MustOverride Public Function Read(ByVal request As HttpRequest) As  
Object()

[JScript] public abstract function Read(request : HttpRequest) : Object[];

*Description*

MimeParameterWriter class (System.Web.Services.Protocols)

ToString

*Description*

MimeParameterWriter

*Example Syntax:*

ToString

[C#] protected MimeParameterWriter();

[C++] protected: MimeParameterWriter();

[VB] Protected Sub New()

[JScript] protected function MimeParameterWriter();

RequestEncoding

ToString

[C#] public virtual Encoding RequestEncoding {get; set;}

[C++] public: \_\_property virtual Encoding\* get\_RequestEncoding();public:

\_\_property virtual void set\_RequestEncoding(Encoding\*);

[VB] Overridable Public Property RequestEncoding As Encoding

[JScript] public function get RequestEncoding() : Encoding;public function set

RequestEncoding(Encoding);

*Description*

UsesWriteRequest

ToString

[C#] public virtual bool UsesWriteRequest {get;}

[C++] public: \_\_property virtual bool get\_UsesWriteRequest();

[VB] Overridable Public ReadOnly Property UsesWriteRequest As Boolean



1 [JScript] public function get UsesWriteRequest() : Boolean;

2  
3 *Description*

4  
5 **GetRequestUrl**

6  
7 [C#] public virtual string GetRequestUrl(string url, object[] parameters);

8 [C++] public: virtual String\* GetRequestUrl(String\* url, Object\* parameters  
9 \_\_gc[]);

10 [VB] Overridable Public Function GetRequestUrl(ByVal url As String, ByVal  
11 parameters() As Object) As String

12 [JScript] public function GetRequestUrl(url : String, parameters : Object[]) :  
13 String;

14  
15 *Description*

16  
17 **InitializeRequest**

18  
19 [C#] public virtual void InitializeRequest(WebRequest request, object[] values);

20 [C++] public: virtual void InitializeRequest(WebRequest\* request, Object\* values  
21 \_\_gc[]);

22 [VB] Overridable Public Sub InitializeRequest(ByVal request As WebRequest,  
23 ByVal values() As Object)

24 [JScript] public function InitializeRequest(request : WebRequest, values :  
25 Object[]);

*Description*

WriteRequest

[C#] public virtual void WriteRequest(Stream requestStream, object[] values);

[C++] public: virtual void WriteRequest(Stream\* requestStream, Object\* values  
\_\_gc[]);

[VB] Overridable Public Sub WriteRequest(ByVal requestStream As Stream,  
ByVal values() As Object)

[JScript] public function WriteRequest(requestStream : Stream, values : Object[]);

*Description*

MimeReturnReader class (System.Web.Services.Protocols)

WriteRequest

*Description*

MimeReturnReader

*Example Syntax:*

WriteRequest

[C#] protected MimeReturnReader();

1 [C++] protected: MimeReturnReader();

2 [VB] Protected Sub New()

3 [JScript] protected function MimeReturnReader();

4       Read

5

6 [C#] public abstract object Read(WebResponse response, Stream

7 responseStream);

8 [C++] public: virtual Object\* Read(WebResponse\* response, Stream\*

9 responseStream) = 0;

10 [VB] MustOverride Public Function Read(ByVal response As WebResponse,

11 ByVal responseStream As Stream) As Object

12 [JScript] public abstract function Read(response : WebResponse, responseStream :

13 Stream) : Object;

14

#### *Description*

NopReturnReader class (System.Web.Services.Protocols)

ToString

#### *Description*

NopReturnReader

*Example Syntax:*

ToString

[C#] public NopReturnReader();

[C++] public: NopReturnReader();

[VB] Public Sub New()

[JScript] public function NopReturnReader();

    GetInitializer

[C#] public override object GetInitializer(LogicalMethodInfo methodInfo);

[C++] public: Object\* GetInitializer(LogicalMethodInfo\* methodInfo);

[VB] Overrides Public Function GetInitializer(ByVal methodInfo As

LogicalMethodInfo) As Object

[JScript] public override function GetInitializer(methodInfo : LogicalMethodInfo)

: Object;

*Description*

    Initialize

[C#] public override void Initialize(object initializer);

[C++] public: void Initialize(Object\* initializer);

[VB] Overrides Public Sub Initialize(ByVal initializer As Object)

[JScript] public override function Initialize(initializer : Object);

*Description*

## Read

```
[C#] public override object Read(WebResponse response, Stream  
responseStream);  
[C++] public: Object* Read(WebResponse* response, Stream* responseStream);  
[VB] Overrides Public Function Read(ByVal response As WebResponse, ByVal  
responseStream As Stream) As Object  
[JScript] public override function Read(response : WebResponse, responseStream  
: Stream) : Object;
```

### *Description*

PatternMatcher class (System.Web.Services.Protocols)

ToString

### *Description*

PatternMatcher

*Example Syntax:*

ToString

```
[C#] public PatternMatcher(Type type);  
[C++] public: PatternMatcher(Type* type);  
[VB] Public Sub New(ByVal type As Type)
```

1 [JScript] public function PatternMatcher(type : Type);

2

3 *Description*

4

5 Match

6

7 [C#] public object Match(string text);

8 [C++] public: Object\* Match(String\* text);

9 [VB] Public Function Match(ByVal text As String) As Object

10 [JScript] public function Match(text : String) : Object;

11

12 *Description*

13

14 SoapClientMessage class (System.Web.Services.Protocols)

15 ToString

16

17

18 *Description*

19 Represents the data in a SOAP request sent or a SOAP response received  
20 by a Web Service client at a specific

21 **System.Web.Services.Protocols.SoapMessageStage** for a Web Service method.

22 Action

23 ToString

24

25 [C#] public override string Action {get;}

1 [C++] public: \_\_property virtual String\* get\_Action();

2 [VB] Overrides Public ReadOnly Property Action As String

3 [JScript] public function get Action() : String;

4  
5 *Description*

6 Gets the SOAPAction HTTP request header field for the SOAP request or  
7 SOAP response.

8 The **System.Web.Services.Protocols.SoapClientMessage.Action** property  
9 is available in any **System.Web.Services.Protocols.SoapMessageStage** .

10 Client

11 ToString

12  
13 [C#] public SoapHttpClientProtocol Client {get;}

14 [C++] public: \_\_property SoapHttpClientProtocol\* get\_Client();

15 [VB] Public ReadOnly Property Client As SoapHttpClientProtocol

16 [JScript] public function get Client() : SoapHttpClientProtocol;

17  
18 *Description*

19 Gets an instance of the client proxy class, which derives from  
20 **System.Web.Services.Protocols.SoapHttpClientProtocol** .

21 The **System.Web.Services.Protocols.SoapClientMessage.Client** property  
22 can be accessed during any **System.Web.Services.Protocols.SoapMessageStage**

23  
24 Content Type

25 Exception

1 Headers

2 MethodInfo

3 ToString

4  
5  
6 *Description*

7 Gets a representation of the method prototype for the Web Service method  
8 for which the SOAP request is intended.

9 Although the  
10 **System.Web.Services.Protocols.SoapClientMessage.MethodInfo** property can  
11 be accessed during any stage, the method information is only available during  
12 **System.Web.Services.Protocols.SoapMessageStage.AfterDeserialize** and  
13 **System.Web.Services.Protocols.SoapMessageStage.BeforeSerialize** .

14 OneWay

15 ToString

16  
17 [C#] public override bool OneWay {get;}

18 [C++] public: \_\_property virtual bool get\_OneWay();

19 [VB] Overrides Public ReadOnly Property OneWay As Boolean

20 [JScript] public function get OneWay() : Boolean;

21  
22 *Description*

23 Gets a value indicating whether the client waits for the server to finish  
24 processing a Web Service method.



1       A **System.Web.Services.Protocols.SoapDocumentMethodAttribute** or  
2 **System.Web.Services.Protocols.SoapRpcMethodAttribute** applied to a Web  
3 Service method or Web Service client can specify whether the Web Service  
4 method is one way or not by setting the  
5 **System.Web.Services.Protocols.SoapDocumentMethodAttribute.OneWay**  
6 property of the attribute.

7       Stage

8       Stream

9       Url

10      ToString

11  
12  
13      *Description*

14       Gets the URL of the Web Service.

15       The **System.Web.Services.Protocols.SoapClientMessage.Url** property  
16 can be accessed during any **System.Web.Services.Protocols.SoapMessageStage**

17  
18      EnsureInStage

19  
20      [C#] protected override void EnsureInStage();

21      [C++] protected: void EnsureInStage();

22      [VB] Overrides Protected Sub EnsureInStage()

23      [JScript] protected override function EnsureInStage();

24  
25      *Description*

1 Asserts that the current  
2 **System.Web.Services.Protocols.SoapMessageStage** stage is a stage where in  
3 parameters are available. If not, an exception is thrown.

4 For Web Service clients, the in parameters are available prior to the SOAP  
5 request serialization process in the  
6 **System.Web.Services.Protocols.SoapMessageStage.BeforeSerialize** stage. The  
7 **System.InvalidOperationException** is thrown if  
8 **System.Web.Services.Protocols.SoapClientMessage.EnsureInStage** method is  
9 invoked in any other **System.Web.Services.Protocols.SoapMessageStage** .

10 **EnsureOutStage**

11  
12 [C#] protected override void EnsureOutStage();  
13 [C++] protected: void EnsureOutStage();  
14 [VB] Overrides Protected Sub EnsureOutStage()  
15 [JScript] protected override function EnsureOutStage();  
16

17 *Description*

18 Asserts that the current  
19 **System.Web.Services.Protocols.SoapMessageStage** stage is a stage where out  
20 parameters are available. If not, an exception is thrown.

21 For Web Service clients, the out parameters are available after the SOAP  
22 response deserialization process in the  
23 **System.Web.Services.Protocols.SoapMessageStage.AfterDeserialize** stage. The  
24 **System.InvalidOperationException** is thrown if  
25

**System.Web.Services.Protocols.SoapClientMessage.EnsureOutStage** method is invoked in any other **System.Web.Services.Protocols.SoapMessageStage** .

**SoapDocumentMethodAttribute** class (System.Web.Services.Protocols)

**ToString**

### *Description*

Applying the optional

**System.Web.Services.Protocols.SoapDocumentMethodAttribute** to a Web Service method alters the format of the SOAP request or response sent to and from a Web Service method.

Web Services Description Language (WSDL) defines two styles for how a Web Service method, which it calls an operation, can be encoded in a SOAP request or a SOAP response: RPC and Document. The Document style refers to encoding the Web Service method as described in an XSD schema. If Document style is used, the Service Description for the Web Service, which is in WSDL, defines XSD schemas for both SOAP requests and SOAP responses to the Web Service method. A Web Service set to the Document encoding style expects clients to pass the XML just as it is defined in these XSD schemas.

**SoapDocumentMethodAttribute**

*Example Syntax:*

**ToString**

[C#] public SoapDocumentMethodAttribute();

[C++] public: SoapDocumentMethodAttribute();

1 [VB] Public Sub New()

2 [JScript] public function SoapDocumentMethodAttribute(); Initializes a new  
3 instance of the

4 **System.Web.Services.Protocols.SoapDocumentMethodAttribute** class.

5  
6 *Description*

7       Initializes a new instance of the

8 **System.Web.Services.Protocols.SoapDocumentMethodAttribute** class.

9       SoapDocumentMethodAttribute

10 *Example Syntax:*

11       ToString

12  
13 [C#] public SoapDocumentMethodAttribute(string action);

14 [C++] public: SoapDocumentMethodAttribute(String\* action);

15 [VB] Public Sub New(ByVal action As String)

16 [JScript] public function SoapDocumentMethodAttribute(action : String);

17  
18 *Description*

19       Initializes a new instance of the

20 **System.Web.Services.Protocols.SoapDocumentMethodAttribute** class setting  
21 the **System.Web.Services.Protocols.SoapDocumentMethodAttribute.Action**

22 property to action. The intent of the SOAP request. Sets the

23 **System.Web.Services.Protocols.SoapDocumentMethodAttribute.Action**  
24 property.

25       Action

ToString

[C#] public string Action {get; set;}

[C++] public: \_\_property String\* get\_Action();public: \_\_property void  
set\_Action(String\*);

[VB] Public Property Action As String

[JScript] public function get Action() : String;public function set Action(String);

*Description*

Gets or sets the intent of the SOAP request.

The

**System.Web.Services.Protocols.SoapDocumentMethodAttribute.Action**

property forms the SOAPAction HTTP Header Field for the HTTP Request.

Binding

ToString

[C#] public string Binding {get; set;}

[C++] public: \_\_property String\* get\_Binding();public: \_\_property void  
set\_Binding(String\*);

[VB] Public Property Binding As String

[JScript] public function get Binding() : String;public function set Binding(String);

*Description*

Gets or sets the binding a Web Service method is implementing a operation  
for.

A binding, as defined by Web Services Description Language(WSDL), is similar to an interface, in that it defines a concrete set of operations. With respect to ASP.NET Web Services, each Web Service method is an operation within a binding. Web Service methods are members of either the default binding for a Web Service or in a binding specified within a **System.Web.Services.WebServiceBindingAttribute** applied to a Web Service. A Web Service can implement multiple bindings, by applying multiple **System.Web.Services.WebServiceBindingAttribute** attributes to a Web Service.

OneWay

ToString

[C#] public bool OneWay {get; set;}

[C++] public: \_\_property bool get\_OneWay();public: \_\_property void set\_OneWay(bool);

[VB] Public Property OneWay As Boolean

[JScript] public function get OneWay() : Boolean;public function set OneWay(Boolean);

### *Description*

Gets or sets whether a Web Service client waits for the Web server to finish processing a Web Service method.

When a Web Service method sets the **System.Web.Services.Protocols.SoapDocumentMethodAttribute.OneWay** property to **true** , the Web Service client does not have to wait for the Web server to finish processing the Web Service method. As soon as the Web server has

1 deserialized the **System.Web.Services.Protocols.SoapServerMessage** , but  
2 before invoking the Web Service method, the server returns an HTTP 202 status  
3 code. A HTTP 202 status code indicates to the client that the Web server has  
4 started processing of the message. Therefore, a Web Service client receives no  
5 acknowledgment that the Web server successfully processed the message.

6 **ParameterStyle**

7 **ToString**

8  
9 [C#] public SoapParameterStyle ParameterStyle {get; set;}

10 [C++] public: \_\_property SoapParameterStyle get\_ParameterStyle();public:

11 \_\_property void set\_ParameterStyle(SoapParameterStyle);

12 [VB] Public Property ParameterStyle As SoapParameterStyle

13 [JScript] public function get ParameterStyle() : SoapParameterStyle;public

14 function set ParameterStyle(SoapParameterStyle);

15  
16 *Description*

17 Gets or sets whether parameters are wrapped within a single element  
18 beneath the **Body** element in the XML portion of a SOAP message.

19 **RequestElementName**

20 **ToString**

21  
22 [C#] public string RequestElementName {get; set;}

23 [C++] public: \_\_property String\* get\_RequestElementName();public: \_\_property

24 void set\_RequestElementName(String\*);

25 [VB] Public Property RequestElementName As String

1 [JScript] public function get RequestElementName() : String;public function set  
2 RequestElementName(String);

3  
4 *Description*

5 Gets or sets the XML element associated with the SOAP request for a Web  
6 Service method.

7 The  
8 **System.Web.Services.Protocols.SoapDocumentMethodAttribute.RequestEle**  
9 **mentName** defines the XML element used to wrap the parameters beneath the  
10 **Body** element of the SOAP request when  
11 **System.Web.Services.Protocols.SoapDocumentMethodAttribute.ParameterSt**  
12 **yle** is **System.Web.Services.Protocols.SoapParameterStyle.Wrapped** . This is  
13 reflected in the XSD schema representing the SOAP request to the Web Service  
14 method within the Web Service's Service Description.

15 RequestNamespace

16 ToString

17  
18 [C#] public string RequestNamespace {get; set;}

19 [C++] public: \_\_property String\* get\_RequestNamespace();public: \_\_property  
20 void set\_RequestNamespace(String\*);

21 [VB] Public Property RequestNamespace As String

22 [JScript] public function get RequestNamespace() : String;public function set  
23 RequestNamespace(String);

24  
25 *Description*



1 Gets or sets the namespace associated with the SOAP request for a Web  
2 Service method.

3 **System.Web.Services.Protocols.SoapDocumentMethodAttribute.Reque**  
4 **stNamespace** is used in the XSD schema for the Web Service method within its  
5 Service Description.

6 ResponseElementName

7 ToString

8  
9 [C#] public string ResponseElementName {get; set;}

10 [C++] public: \_\_property String\* get\_ResponseElementName();public: \_\_property  
11 void set\_ResponseElementName(String\*);

12 [VB] Public Property ResponseElementName As String

13 [JScript] public function get ResponseElementName() : String;public function set  
14 ResponseElementName(String);

15  
16 *Description*

17 Gets or sets the XML element associated with the SOAP response for a  
18 Web Service method.

19 The

20 **System.Web.Services.Protocols.SoapDocumentMethodAttribute.ResponseEle**  
21 **mentName** defines the XML element used to wrap the parameters beneath the  
22 **Body** element of the SOAP response when

23 **System.Web.Services.Protocols.SoapDocumentMethodAttribute.ParameterSt**  
24 **yle** is **System.Web.Services.Protocols.SoapParameterStyle.Wrapped** . This is  
25

1 reflected in the XSD schema representing the SOAP response to the Web Service  
2 method within the Web Service's Service Description.

3       ResponseNamespace

4       ToString

6 [C#] public string ResponseNamespace {get; set;}

7 [C++] public: \_\_property String\* get\_ResponseNamespace();public: \_\_property

8 void set\_ResponseNamespace(String\*);

9 [VB] Public Property ResponseNamespace As String

10 [JScript] public function get ResponseNamespace() : String;public function set

11 ResponseNamespace(String);

13 *Description*

14       Gets or sets the XML namespace associated with the SOAP response for a  
15 Web Service method.

16       The

17 **System.Web.Services.Protocols.SoapDocumentMethodAttribute.ResponseNa**

18 **mespace** property is used in the XSD schema for the Web Service method in its

19 Service Description.

20       TypeId

21       Use

22       ToString

25 *Description*

1 Gets or sets the parameter encoding for a Web Service method within the  
2 XML portion of a SOAP message.

3 The Web Services Description Language (WSDL) defines two parameter  
4 encoding styles: **System.Web.Services.Description.SoapBindingUse.Encoded**  
5 and **System.Web.Services.Description.SoapBindingUse.Literal** .

6 SoapDocumentServiceAttribute class (System.Web.Services.Protocols)

7 ToString

8  
9  
10 *Description*

11 Applying the optional  
12 **System.Web.Services.Protocols.SoapDocumentServiceAttribute** to a Web  
13 Service sets the default format of SOAP requests and responses sent to and from  
14 Web Service methods within the Web Service.

15 The **System.Web.Services.Protocols.SoapDocumentServiceAttribute**  
16 allows you to set the default encoding styles for Web Service methods within a  
17 Web Service. If an individual Web Service method needs to change these defaults,  
18 apply a **System.Web.Services.Protocols.SoapDocumentMethodAttribute** to  
19 that Web Service method.

20 SoapDocumentServiceAttribute

21 *Example Syntax:*

22 ToString

23  
24 [C#] public SoapDocumentServiceAttribute();

25 [C++] public: SoapDocumentServiceAttribute();

1 [VB] Public Sub New()

2 [JScript] public function SoapDocumentServiceAttribute(); Initializes a new  
3 instance of the

4 **System.Web.Services.Protocols.SoapDocumentServiceAttribute** class.

5  
6 *Description*

7       Initializes a new instance of the  
8 **System.Web.Services.Protocols.SoapDocumentServiceAttribute** class setting  
9 all properties to their defaults.

10       SoapDocumentServiceAttribute

11       *Example Syntax:*

12       ToString

13  
14 [C#] public SoapDocumentServiceAttribute(SoapBindingUse use);

15 [C++] public: SoapDocumentServiceAttribute(SoapBindingUse use);

16 [VB] Public Sub New(ByVal use As SoapBindingUse)

17 [JScript] public function SoapDocumentServiceAttribute(use : SoapBindingUse);

18  
19 *Description*

20       Initializes a new instance of the  
21 **System.Web.Services.Protocols.SoapDocumentServiceAttribute** class setting  
22 the parameter encoding. The parameter encoding of the Web Service. Sets the  
23 **System.Web.Services.Protocols.SoapDocumentServiceAttribute.Use** property.

24       SoapDocumentServiceAttribute

25       *Example Syntax:*

ToString

```
[C#] public SoapDocumentServiceAttribute(SoapBindingUse use,  
SoapParameterStyle paramStyle);  
[C++] public: SoapDocumentServiceAttribute(SoapBindingUse use,  
SoapParameterStyle paramStyle);  
[VB] Public Sub New(ByVal use As SoapBindingUse, ByVal paramStyle As  
SoapParameterStyle)  
[JScript] public function SoapDocumentServiceAttribute(use : SoapBindingUse,  
paramStyle : SoapParameterStyle);
```

### *Description*

Initializes a new instance of the **System.Web.Services.Protocols.SoapDocumentServiceAttribute** class setting the parameter encoding and whether parameters are wrapped within a single element under the **Body** element within the XML portion of SOAP requests and responses. The parameter encoding style. Sets the **System.Web.Services.Protocols.SoapDocumentServiceAttribute.Use** property. Sets whether parameters are wrapped withn a single XML element under the **Body** element within the XML portion of SOAP requests and responses to Web Service methods within the Web Service. Sets the **System.Web.Services.Protocols.SoapDocumentServiceAttribute.ParameterStyle** property.

ParameterStyle

ToString

1  
2 [C#] public SoapParameterStyle ParameterStyle {get; set;}

3 [C++] public: \_\_property SoapParameterStyle get\_ParameterStyle();public:

4 \_\_property void set\_ParameterStyle(SoapParameterStyle);

5 [VB] Public Property ParameterStyle As SoapParameterStyle

6 [JScript] public function get ParameterStyle() : SoapParameterStyle;public

7 function set ParameterStyle(SoapParameterStyle);

8  
9 *Description*

10 Gets or sets the default for Web Service methods within the Web Service  
11 whether parameters are wrapped within a single element beneath the **Body**  
12 element in the XML portion of a SOAP message.

13 RoutingStyle

14 ToString

15  
16 [C#] public SoapServiceRoutingStyle RoutingStyle {get; set;}

17 [C++] public: \_\_property SoapServiceRoutingStyle get\_RoutingStyle();public:

18 \_\_property void set\_RoutingStyle(SoapServiceRoutingStyle);

19 [VB] Public Property RoutingStyle As SoapServiceRoutingStyle

20 [JScript] public function get RoutingStyle() : SoapServiceRoutingStyle;public

21 function set RoutingStyle(SoapServiceRoutingStyle);

22  
23 *Description*

24  
25 TypeId

1 Use

2 ToString

3  
4  
5 *Description*

6 Gets or sets the default parameter encoding for a Web Service.

7 The Web Services Description Language (WSDL) defines two parameter  
8 encoding styles: **System.Web.Services.Description.SoapBindingUse.Encoded**  
9 and **System.Web.Services.Description.SoapBindingUse.Literal** .

10 SoapException class (System.Web.Services.Protocols)

11 ToString

12  
13  
14 *Description*

15 The exception that is thrown when a Web Service method is called over  
16 Simple Object Access Protocol (SOAP) and an exception occurs.

17 The **System.Web.Services.Protocols.SoapException** can either be thrown  
18 by the common language runtime or by a Web Service method. The common  
19 language runtime can throw a **System.Web.Services.Protocols.SoapException** if  
20 a response to a request is not formatted correctly. Web Service methods can  
21 generate a **System.Web.Services.Protocols.SoapException** by simply throwing  
22 an exception within the Web Service method. If the client accessed the method  
23 over SOAP, the exception is caught on the server and wrapped inside a new  
24 **System.Web.Services.Protocols.SoapException** . The SoapException thrown has

the following property values: Property Value **System.Exception.Message** The **System.Exception.Message** property of the original exception.

ToString

[C#] public static readonly XmlQualifiedName ClientFaultCode;

[C++] public: static XmlQualifiedName\* ClientFaultCode;

[VB] Public Shared ReadOnly ClientFaultCode As XmlQualifiedName

[JScript] public static var ClientFaultCode : XmlQualifiedName;

#### *Description*

Specifies a SOAP fault code representing a client call was not formatted correctly or did not contain the appropriate information.

An example of when a **System.Web.Services.Protocols.SoapException.ClientFaultCode** can be generated, is when a client call lacks proper authentication or payment information. It is generally an indication that the client call should not be resent without change.

ToString

[C#] public static readonly XmlQualifiedName DetailElementName;

[C++] public: static XmlQualifiedName\* DetailElementName;

[VB] Public Shared ReadOnly DetailElementName As XmlQualifiedName

[JScript] public static var DetailElementName : XmlQualifiedName;

#### *Description*



1 Gets an **System.Xml.XmlQualifiedName** representing the  
2 **System.Web.Services.Protocols.SoapException.Detail** element of a SOAP Fault  
3 code.

4 In building an **System.Xml.XmlNode** for the  
5 **System.Web.Services.Protocols.SoapException.Detail** property, the  
6 **System.Xml.XmlQualifiedName.Name** and  
7 **System.Xml.XmlQualifiedName.Namespace** properties of  
8 **System.Web.Services.Protocols.SoapException.DetailElementName** can be  
9 used to ensure consistency with the SOAP specification.

10 ToString

11  
12 [C#] public static readonly XmlQualifiedName MustUnderstandFaultCode;  
13 [C++] public: static XmlQualifiedName\* MustUnderstandFaultCode;  
14 [VB] Public Shared ReadOnly MustUnderstandFaultCode As XmlQualifiedName  
15 [JScript] public static var MustUnderstandFaultCode : XmlQualifiedName;  
16

17 *Description*

18 A SOAP Fault Code representing a SOAP element marked with the  
19 MustUnderstand attribute was not processed.

20 Not all SOAP elements require processing by the server. However, if a  
21 SOAP element is marked with the MustUnderstand attribute equal to 1, processing  
22 is required. Failure to process the element, generates a  
23 **System.Web.Services.Protocols.SoapException** with a  
24 **System.Web.Services.Protocols.SoapException.MustUnderstandFaultCode** .

25 ToString

```

1  [C#] public static readonly XmlQualifiedName ServerFaultCode;
2
3  [C++] public: static XmlQualifiedName* ServerFaultCode;
4
5  [VB] Public Shared ReadOnly ServerFaultCode As XmlQualifiedName
6
7  [JScript] public static var ServerFaultCode : XmlQualifiedName;
8

```

### *Description*

Specifies a SOAP fault code representing an error occurred during the processing of a client call on the server, where the problem was not due to the message contents.

A **System.Web.Services.Protocols.SoapException.ServerFaultCode** might occur for example, if a server being called, couldn't respond to a request due to network problems. Typically, with this type of exception, the client call may succeed later.

### *ToString*

```

17 [C#] public static readonly XmlQualifiedName VersionMismatchFaultCode;
18
19 [C++] public: static XmlQualifiedName* VersionMismatchFaultCode;
20
21 [VB] Public Shared ReadOnly VersionMismatchFaultCode As
22 XmlQualifiedName
23
24 [JScript] public static var VersionMismatchFaultCode : XmlQualifiedName;
25

```

### *Description*

A SOAP fault code representing an invalid namespace for a SOAP Envelope was found during the processing of the SOAP message.

1 A SOAP Envelope is the top-level element of a XML document  
2 representing a SOAP message.

3 SoapException

4 *Example Syntax:*

5 ToString

6  
7 [C#] public SoapException(string message, XmlQualifiedName code);

8 [C++] public: SoapException(String\* message, XmlQualifiedName\* code);

9 [VB] Public Sub New(ByVal message As String, ByVal code As

10 XmlQualifiedName)

11 [JScript] public function SoapException(message : String, code :

12 XmlQualifiedName);

13  
14 *Description*

15 Initializes a new instance of the  
16 **System.Web.Services.Protocols.SoapException** class that sets the exception  
17 message and exception code. A message that identifies the reason the exception  
18 occurred. Sets the **System.Exception.Message** property. The type of error that  
19 occurred. Sets the **System.Web.Services.Protocols.SoapException.Code**  
20 property.

21 SoapException

22 *Example Syntax:*

23 ToString

24  
25 [C#] public SoapException(string message, XmlQualifiedName code, Exception

```

1 innerException);
2 [C++] public: SoapException(String* message, XmlQualifiedName* code,
3 Exception* innerException);
4 [VB] Public Sub New(ByVal message As String, ByVal code As
5 XmlQualifiedName, ByVal innerException As Exception)
6 [JScript] public function SoapException(message : String, code :
7 XmlQualifiedName, innerException : Exception);
8

```

### *Description*

Initializes a new instance of the **System.Web.Services.Protocols.SoapException** class that sets the exception message, exception code and a reference to the root cause of the exception. A message that identifies the reason the exception occurred. Sets the **System.Exception.Message** property. The type of error that occurred. Sets the **System.Web.Services.Protocols.SoapException.Code** property. A reference to the root cause of the exception. Sets the **System.Exception.InnerException** property.

**SoapException**

*Example Syntax:*

**ToString**

```

22 [C#] public SoapException(string message, XmlQualifiedName code, string
23 actor);
24 [C++] public: SoapException(String* message, XmlQualifiedName* code,
25 String* actor);

```

```

1 [VB] Public Sub New(ByVal message As String, ByVal code As
2 XmlQualifiedNames, ByVal actor As String)
3 [JScript] public function SoapException(message : String, code :
4 XmlQualifiedNames, actor : String); Initializes a new instance of the
5 System.Web.Services.Protocols.SoapException class.

```

### *Description*

Initializes a new instance of the **System.Web.Services.Protocols.SoapException** class that sets the exception message, exception code and the piece of code that caused the exception. A message that identifies the reason the exception occurred. Sets the **System.Exception.Message** property. The type of error that occurred. Sets the **System.Web.Services.Protocols.SoapException.Code** property. The piece of code that caused the exception. Typically, a URL to a Web Service method. Sets the **System.Web.Services.Protocols.SoapException.Actor** property.

SoapException

*Example Syntax:*

ToString

```

20 [C#] public SoapException(string message, XmlQualifiedNames code, string actor,
21 Exception innerException);
22 [C++] public: SoapException(String* message, XmlQualifiedNames* code,
23 String* actor, Exception* innerException);
24 [VB] Public Sub New(ByVal message As String, ByVal code As
25 XmlQualifiedNames, ByVal actor As String, ByVal innerException As Exception)

```

1 [JScript] public function SoapException(message : String, code :  
2 XmlQualifiedName, actor : String, innerException : Exception);

3  
4 *Description*

5       Initializes a new instance of the  
6 **System.Web.Services.Protocols.SoapException** class that sets the exception  
7 message, exception code, the piece of code that caused the exception and a  
8 reference to the root cause of the exception. A message that identifies the reason  
9 the exception occurred. Sets the **System.Exception.Message** property. The type of  
10 error that occurred. Sets the  
11 **System.Web.Services.Protocols.SoapException.Code** property. The piece of  
12 code that caused the exception. Typically, a URL to a Web Service method. Sets  
13 the **System.Web.Services.Protocols.SoapException.Actor** property. A reference  
14 to the root cause of an exception. Sets the **System.Exception.InnerException**  
15 property.

16       SoapException

17       *Example Syntax:*

18       ToString

19  
20 [C#] public SoapException(string message, XmlQualifiedName code, string actor,  
21 XmlNode detail);

22 [C++] public: SoapException(String\* message, XmlQualifiedName\* code,  
23 String\* actor, XmlNode\* detail);

24 [VB] Public Sub New(ByVal message As String, ByVal code As  
25 XmlQualifiedName, ByVal actor As String, ByVal detail As XmlNode)

1 [JScript] public function SoapException(message : String, code :  
2 XmlQualifiedName, actor : String, detail : XmlNode);

3  
4 *Description*

5       Initializes a new instance of the  
6 **System.Web.Services.Protocols.SoapException** class that sets the exception  
7 message, exception code, piece of code that caused the exception and application  
8 specific exception information. A message that identifies the reason the exception  
9 occurred. Sets the **System.Exception.Message** property. The type of error that  
10 occurred. Sets the **System.Web.Services.Protocols.SoapException.Code**  
11 property. The piece of code that caused the exception. Typically, a URL to a Web  
12 Service method. Sets the **System.Web.Services.Protocols.SoapException.Actor**  
13 property. The application specific exception information. Sets the  
14 **System.Web.Services.Protocols.SoapException.Detail** property.

15       SoapException

16       *Example Syntax:*

17       ToString

18  
19 [C#] public SoapException(string message, XmlQualifiedName code, string actor,  
20 XmlNode detail, Exception innerException);

21 [C++] public: SoapException(String\* message, XmlQualifiedName\* code,  
22 String\* actor, XmlNode\* detail, Exception\* innerException);

23 [VB] Public Sub New(ByVal message As String, ByVal code As  
24 XmlQualifiedName, ByVal actor As String, ByVal detail As XmlNode, ByVal  
25 innerException As Exception)

[JScript] public function SoapException(message : String, code :  
XmlQualifiedName, actor : String, detail : XmlNode, innerException : Exception);

#### *Description*

Initializes a new instance of the  
**System.Web.Services.Protocols.SoapException** class that sets the exception  
message, exception code, piece of code that caused the exception, application  
specific exception information and a reference to the root cause of the exception.  
A message that identifies the reason the exception occurred. Sets the  
**System.Exception.Message** property. The type of error that occurred. Sets the  
**System.Web.Services.Protocols.SoapException.Code** property. The piece of  
code that caused the exception. Typically, a URL to a Web Service method. Sets  
the **System.Web.Services.Protocols.SoapException.Actor** property. The  
application specific exception information. Sets the  
**System.Web.Services.Protocols.SoapException.Detail** property. A reference to  
the root cause of the exception. Sets the **System.Exception.InnerException**  
property.

SoapException

*Example Syntax:*

ToString

[C#] public SoapException(string message, XmlQualifiedName code, string actor,  
XmlNode detail, XmlNode[] otherElements);  
[C++] public: SoapException(String\* message, XmlQualifiedName\* code,  
String\* actor, XmlNode\* detail, XmlNode\* otherElements[]);



```

1 [VB] Public Sub New(ByVal message As String, ByVal code As
2 XmlQualifiedName, ByVal actor As String, ByVal detail As XmlNode, ByVal
3 otherElements() As XmlNode)
4 [JScript] public function SoapException(message : String, code :
5 XmlQualifiedName, actor : String, detail : XmlNode, otherElements :
6 XmlNode[]);

```

### *Description*

Initializes a new instance of the **System.Web.Services.Protocols.SoapException** class that sets the exception message, exception code, piece of code that caused the exception, application specific exception information and optional exception information. A message that identifies the reason the exception occurred. Sets the **System.Exception.Message** property. The type of error that occurred. Sets the **System.Web.Services.Protocols.SoapException.Code** property. The piece of code that caused the exception. Typically, a URL to a Web Service method. Sets the **System.Web.Services.Protocols.SoapException.Actor** property. The application specific exception information. Sets the **System.Web.Services.Protocols.SoapException.Detail** property. Optional exception information. Sets the **System.Web.Services.Protocols.SoapException.OtherElements** property.

SoapException

*Example Syntax:*

ToString

```

1
2 [C#] public SoapException(string message, XmlQualifiedName code, string actor,
3 XmlNode detail, XmlNode[] otherElements, Exception innerException);
4 [C++] public: SoapException(String* message, XmlQualifiedName* code,
5 String* actor, XmlNode* detail, XmlNode* otherElements[], Exception*
6 innerException);
7 [VB] Public Sub New(ByVal message As String, ByVal code As
8 XmlQualifiedName, ByVal actor As String, ByVal detail As XmlNode, ByVal
9 otherElements() As XmlNode, ByVal innerException As Exception)
10 [JScript] public function SoapException(message : String, code :
11 XmlQualifiedName, actor : String, detail : XmlNode, otherElements : XmlNode[],
12 innerException : Exception);
13

```

#### *Description*

Initializes a new instance of the **System.Web.Services.Protocols.SoapException** class that sets the exception message, exception code, piece of code that caused the exception, application specific exception information, optional exception information, and a reference to the root cause of the exception. A message that identifies the reason the exception occurred. Sets the **System.Exception.Message** property. The type of error that occurred. Sets the **System.Web.Services.Protocols.SoapException.Code** property. The piece of code that caused the exception. Typically, a URL to a Web Service method. Sets the **System.Web.Services.Protocols.SoapException.Actor** property. The application specific exception information. Sets the **System.Web.Services.Protocols.SoapException.Detail** property. Optional

1 exception information. Sets the

2 **System.Web.Services.Protocols.SoapException.OtherElements** property. A

3 reference to the root cause of the exception. Sets the

4 **System.Exception.InnerException** property.

5 Actor

6 ToString

7  
8 [C#] public string Actor {get;}

9 [C++] public: \_\_property String\* get\_Actor();

10 [VB] Public ReadOnly Property Actor As String

11 [JScript] public function get Actor() : String;

12  
13 *Description*

14 Gets the piece of code that caused the exception.

15 **System.Web.Services.Protocols.SoapException.Actor** can only be set  
16 using one of the constructors that accept an

17 **System.Web.Services.Protocols.SoapException.Actor** argument.

18 Code

19 ToString

20  
21 [C#] public XmlQualifiedName Code {get;}

22 [C++] public: \_\_property XmlQualifiedName\* get\_Code();

23 [VB] Public ReadOnly Property Code As XmlQualifiedName

24 [JScript] public function get Code() : XmlQualifiedName;

1  
2 *Description*

3 Gets the type of SOAP fault code.

4 **System.Web.Services.Protocols.SoapException.Code** can only be set  
5 when creating a new instance of the  
6 **System.Web.Services.Protocols.SoapException** class.

7 Detail

8 ToString

9  
10 [C#] public XmlNode Detail {get;}

11 [C++] public: \_\_property XmlNode\* get\_Detail();

12 [VB] Public ReadOnly Property Detail As XmlNode

13 [JScript] public function get Detail() : XmlNode;

14  
15 *Description*

16 Gets an **System.Xml.XmlNode** representing the application specific error  
17 information.

18 **System.Web.Services.Protocols.SoapException.Detail** can be set using  
19 one of the class constructors that accept a  
20 **System.Web.Services.Protocols.SoapException.Detail** value.

21 HelpLink

22 HResult

23 InnerException

24 Message

25 OtherElements

ToString

*Description*

Gets an array of **System.Xml.XmlNode** with the optional error information.

**System.Web.Services.Protocols.SoapException.OtherElements** can be set using one of the class constructors that accept an argument for **System.Web.Services.Protocols.SoapException.OtherElements**.

Source

StackTrace

TargetSite

SoapExtension class (System.Web.Services.Protocols)

ToString

*Description*

Represents the base class for SOAP extensions in ASP.NET Web Services.

ASP.NET Web Services allows a SOAP-related infrastructure to be built by means of an extensibility mechanism. The ASP.NET SOAP extension architecture revolves around an extension that can inspect or modify a message at specific stages in message processing on either the client or the server.

SoapExtension

*Example Syntax:*

ToString

1 [C#] protected SoapExtension();

2 [C++] protected: SoapExtension();

3 [VB] Protected Sub New()

4 [JScript] protected function SoapExtension();

5 ChainStream

6 [C#] public virtual Stream ChainStream(Stream stream);

7 [C++] public: virtual Stream\* ChainStream(Stream\* stream);

8 [VB] Overridable Public Function ChainStream(ByVal stream As Stream) As

9 Stream

10 [JScript] public function ChainStream(stream : Stream) : Stream;

## 11 Description

12 When overridden in a derived class, allows a SOAP extension access to the  
13 memory buffer containing the SOAP request or response.

14 *Return Value:* A **System.IO.Stream** representing a new memory buffer that this  
15 SOAP extension can modify.

16 **System.Web.Services.Protocols.SoapExtension.ChainStream(System.I**  
17 **O.Stream)** ensures that SOAP extensions with the highest priority can modify the  
18 actual data closest to the SOAP message sent or returned over the wire. For  
19 instance, you probably would not want a compression SOAP extension to  
20 compress the data prior to an encryption SOAP extension encrypted the data.  
21 Doing so might lead to an inefficiently compressed data packet sent over the wire.  
22 A memory buffer containing the SOAP request or response.

## GetInitializer

```
[C#] public abstract object GetInitializer(Type serviceType);  
[C++] public: virtual Object* GetInitializer(Type* serviceType) = 0;  
[VB] MustOverride Public Function GetInitializer(ByVal serviceType As Type)  
As Object  
[JScript] public abstract function GetInitializer(serviceType : Type) : Object;
```

### *Description*

When overridden in a derived class, allows a SOAP extension to initialize data specific to a class implementing a Web Service at a one-time performance cost.

*Return Value:* The **System.Object** that the SOAP extension initializes to be cached.

The overload of **System.Web.Services.Protocols.SoapExtension.GetInitializer** that gets called by the ASP.NET Web Services infrastructure depends on how the SOAP extension was specified. There are two methods for specifying a SOAP extension: apply a custom attribute, deriving from **System.Web.Services.Protocols.SoapExtensionAttribute**, to the individual Web Service method or add a reference in either the web.config or app.config configuration files. If you add a reference to one of the configuration files, the SOAP extension runs for all Web Services within the scope of that configuration file. The ASP.NET Web Services infrastructure invokes the **System.Web.Services.Protocols.GetInitializer** overload that passes in a

1 **System.Type** ; otherwise the ASP.NET Web Services infrastructure invokes the  
2 **System.Web.Services.Protocols.SoapExtension.GetInitializer(System.Web.Se**  
3 **rvices.Protocols.LogicalMethodInfo, System.Web.Services.Protocols.SoapExte**  
4 **nsionAttribute)** that passes in a  
5 **System.Web.Services.Protocols.LogicalMethodInfo** and a  
6 **System.Web.Services.Protocols.SoapExtensionAttribute** . The type of the class  
7 implementing the Web Service that a SOAP extension is applied to.

8       **GetInitializer**

9  
10 [C#] public abstract object GetInitializer(LogicalMethodInfo methodInfo,  
11 SoapExtensionAttribute attribute);  
12 [C++] public: virtual Object\* GetInitializer(LogicalMethodInfo\* methodInfo,  
13 SoapExtensionAttribute\* attribute) = 0;  
14 [VB] MustOverride Public Function GetInitializer(ByVal methodInfo As  
15 LogicalMethodInfo, ByVal attribute As SoapExtensionAttribute) As Object  
16 [JScript] public abstract function GetInitializer(methodInfo : LogicalMethodInfo,  
17 attribute : SoapExtensionAttribute) : Object; When overridden in a derived class,  
18 allows a SOAP extension to initialize data specific to a Web Service method at a  
19 one-time performance cost.  
20

### 21 *Description*

22       When overridden in a derived class, allows a SOAP extension to initialize  
23 data specific to a Web Service method using an attribute applied to the Web  
24 Service method at a one-time performance cost.  
25



1 *Return Value:* The **System.Object** that the SOAP extension initializes to be  
2 cached.

3 If the SOAP extension is configured using a configuration file see the  
4 **System.Web.Services.Protocols.SoapExtension.GetInitialzier** overload that  
5 accepts a **System.Type** . A **System.Web.Services.Protocols.LogicalMethodInfo**  
6 object representing the specific function prototype for the Web Service method the  
7 SOAP extension is applied to. The instance of the  
8 **System.Web.Services.Protocols.SoapExtensionAttribute** applied to the Web  
9 Service method.

#### 10 Initialize

11  
12 [C#] public abstract void Initialize(object initializer);  
13 [C++] public: virtual void Initialize(Object\* initializer) = 0;  
14 [VB] MustOverride Public Sub Initialize(ByVal initializer As Object)  
15 [JScript] public abstract function Initialize(initializer : Object);  
16

#### 17 *Description*

18 When overridden in a derived class, allows a SOAP extension to initialize  
19 itself using the data cached in the  
20 **System.Web.Services.Protocols.SoapExtension.GetInitializer(System.Web.Se**  
21 **rvice**s.Protocols.LogicalMethodInfo, System.Web.Services.Protocols.SoapExte  
22 **nsionAttribute)** method.

23 A SOAP extension has three opportunities to initialize data and they all  
24 have different purposes: Class constructor - The class constructor is called every  
25 time a SOAP extension is instantiated and is typically used to initialize member

variables. The **System.Object** returned from

**System.Web.Services.Protocols.SoapExtension.GetInitializer(System.Web.Services.Protocols.LogicalMethodInfo, System.Web.Services.Protocols.SoapExtensionAttribute)** cached by ASP.NET Web Services.

**ProcessMessage**

[C#] public abstract void ProcessMessage(SoapMessage message);

[C++] public: virtual void ProcessMessage(SoapMessage\* message) = 0;

[VB] MustOverride Public Sub ProcessMessage(ByVal message As SoapMessage)

[JScript] public abstract function ProcessMessage(message : SoapMessage);

### *Description*

When overridden in a derived class, allows a SOAP extension to receive a **System.Web.Services.Protocols.SoapMessage** to process at each **System.Web.Services.Protocols.SoapMessageStage**.

**System.Web.Services.Protocols.SoapExtension.ProcessMessage(System.Web.Services.Protocols.SoapMessage)** is called at all **System.Web.Services.Protocols.SoapMessageStage** stages for SOAP extensions applied to both ASP.NET Web Service clients and ASP.NET Web Services. At each **System.Web.Services.Protocols.SoapMessageStage**, an instance of a class deriving from **System.Web.Services.Protocols.SoapMessage** is passed to **System.Web.Services.Protocols.SoapExtension.ProcessMessage(System.Web.Services.Protocols.SoapMessage)**. If the SOAP extension is running on the Web Service client, then a **System.Web.Services.Protocols.SoapClientMessage** object

is passed into

**System.Web.Services.Protocols.SoapExtension.ProcessMessage(System.Web.Services.Protocols.SoapMessage)** ; otherwise a **System.Web.Services.Protocols.SoapServerMessage** object is passed in. The **System.Web.Services.Protocols.SoapMessage** to process.

**SoapExtensionAttribute** class (System.Web.Services.Protocols)

**ToString**

### *Description*

When overridden in a derived class, specifies the SOAP extension should be applied to a Web Service method.

ASP.NET Web Services enables applying SOAP extensions to a Web Service method by applying an attribute. When a custom extension attribute is added to a Web Service method or a proxy class client, ASP.NET Web Services invokes the associated extension at the appropriate time. An extension attribute is a custom attribute class deriving from

**System.Web.Services.Protocols.SoapExtensionAttribute** . Derived attributes must override the

**System.Web.Services.Protocols.SoapExtensionAttribute.ExtensionType** property to return the type of extension that is associated with the attribute.

**SoapExtensionAttribute**

### *Example Syntax:*

**ToString**

1  
2 [C#] protected SoapExtensionAttribute();

3 [C++] protected: SoapExtensionAttribute();

4 [VB] Protected Sub New()

5 [JScript] protected function SoapExtensionAttribute();

6       ExtensionType

7       ToString

8  
9 [C#] public abstract Type ExtensionType {get;}

10 [C++] public: \_\_property virtual Type\* get\_ExtensionType() = 0;

11 [VB] MustOverride Public ReadOnly Property ExtensionType As Type

12 [JScript] public abstract function get ExtensionType() : Type;

13  
14 *Description*

15       When overridden in a derived class, gets the **System.Type** of the SOAP  
16 extension.

17       Derived classes must override the  
18 **System.Web.Services.Protocols.SoapExtensionAttribute.ExtensionType**  
19 property to return the **System.Type** of SOAP extension.

20       Priority

21       ToString

22  
23 [C#] public abstract int Priority {get; set;}

24 [C++] public: \_\_property virtual int get\_Priority() = 0;public: \_\_property virtual

25 void set\_Priority(int) = 0;

1 [VB] MustOverride Public Property Priority As Integer

2 [JScript] public abstract function get Priority() : int;public abstract function set  
3 Priority(int);

4  
5 *Description*

6 When overridden in a derived class, gets or set the priority of the SOAP  
7 extension.

8 TypeId

9 SoapHeader class (System.Web.Services.Protocols)

10 ToString

11  
12  
13 *Description*

14 When overridden in a derived class, represents the content of a SOAP  
15 header.

16 SOAP headers offer a method for passing data to and from a Web Service  
17 method not directly related to a Web Service method's primary functionality. For  
18 instance, a Web Service may contain several Web Service methods that each  
19 require a custom authentication scheme. Instead of adding parameters to each Web  
20 Service method for the custom authentication scheme, a

21 **System.Web.Services.Protocols.SoapHeaderAttribute** , referring to a class  
22 deriving from **System.Web.Services.Protocols.SoapHeader** , can be applied to  
23 each Web Service method. The implementation for the class deriving from  
24 **System.Web.Services.Protocols.SoapHeader** handles the custom authentication  
25 scheme. In this manner, the Web Service method implements only the

1 functionality specific to it and adds additional functionality through the use of a  
2 SOAP header.

3 SoapHeader

4 *Example Syntax:*

5 ToString

6  
7 [C#] protected SoapHeader();

8 [C++] protected: SoapHeader();

9 [VB] Protected Sub New()

10 [JScript] protected function SoapHeader();

11 Actor

12 ToString

13  
14 [C#] public string Actor {get; set;}

15 [C++] public: \_\_property String\* get\_Actor();public: \_\_property void

16 set\_Actor(String\*);

17 [VB] Public Property Actor As String

18 [JScript] public function get Actor() : String;public function set Actor(String);

19  
20 *Description*

21 Gets or sets the recipient of the SOAP header.

22 According to the SOAP specification, the recipient (commonly called the  
23 SOAP actor attribute) does not have to be the same for the **Header** and **Body**  
24 elements of a SOAP message. For a Web Service method, the required **Body**  
25 element of a SOAP message represents the parameters and return values of the

1 Web Service method. If a **Header** element exists in the SOAP message, it  
2 represents additional data that can be sent to and from the Web Service method.  
3 The recipient of that data, known as the **Actor** , can be a different URI than the  
4 URI for the Web Service method.

5 DidUnderstand

6 ToString

7  
8 [C#] public bool DidUnderstand {get; set;}

9 [C++] public: \_\_property bool get\_DidUnderstand();public: \_\_property void  
10 set\_DidUnderstand(bool);

11 [VB] Public Property DidUnderstand As Boolean

12 [JScript] public function get DidUnderstand() : Boolean;public function set  
13 DidUnderstand(Boolean);

14  
15 *Description*

16 Gets or sets a value indicating whether a Web Service method properly  
17 processed a SOAP header.

18 For SOAP headers defined by a Web Service, ASP.NET Web Services  
19 assumes the Web Service method properly processed the SOAP header by setting  
20 the initial value of

21 **System.Web.Services.Protocols.SoapHeader.DidUnderstand** to **true** . For  
22 SOAP headers not defined by the Web Service, the initial value is **false** . If  
23 ASP.NET Web Services detects SOAP headers passed to a Web Service method  
24 with **System.Web.Services.Protocols.SoapHeader.DidUnderstand** set to **false**  
25 after the method returns, a

**System.Web.Services.Protocols.SoapHeaderException** is thrown back to the Web Service client instead of the results from the Web Service method.

EncodedMustUnderstand

ToString

[C#] public string EncodedMustUnderstand {get; set;}

[C++] public: \_\_property String\* get\_EncodedMustUnderstand();public:

\_\_property void set\_EncodedMustUnderstand(String\*);

[VB] Public Property EncodedMustUnderstand As String

[JScript] public function get EncodedMustUnderstand() : String;public function

set EncodedMustUnderstand(String);

### *Description*

Gets or sets the value of the **mustUnderstand** attribute for the SOAP header.

Valid values for the

**System.Web.Services.Protocols.SoapHeader.EncodedMustUnderstand**

property are 0 and 1, whereas setting the property to false or true equates to setting the property to 0 or 1, respectively.

MustUnderstand

ToString

[C#] public bool MustUnderstand {get; set;}

[C++] public: \_\_property bool get\_MustUnderstand();public: \_\_property void

set\_MustUnderstand(bool);



1 [VB] Public Property MustUnderstand As Boolean

2 [JScript] public function get MustUnderstand() : Boolean;public function set

3 MustUnderstand(Boolean);

4  
5 *Description*

6 Gets or sets a value indicating whether the  
7 **System.Web.Services.Protocols.SoapHeader** must be understood.

8 When a Web Service client adds a SOAP header to a Web Service method  
9 call with the **System.Web.Services.Protocols.SoapHeader.MustUnderstand**  
10 property set to **true** , the Web Service method must set the  
11 **System.Web.Services.Protocols.SoapHeader.DidUnderstand** property to **true** ;  
12 otherwise, a **System.Web.Services.Protocols.SoapHeaderException** is thrown  
13 back to the Web Service client by ASP.NET Web Services.

14 SoapHeaderAttribute class (System.Web.Services.Protocols)

15 ToString

16  
17  
18 *Description*

19 This attribute is applied to a Web Service or a Web Service client to specify  
20 a SOAP Header the Web Service or Web Service client wants to process. This  
21 class cannot be inherited.

22 The basic steps to receiving and processing a SOAP header are: Create a  
23 class deriving from **System.Web.Services.Protocols.SoapHeader** representing  
24 the data passed in the SOAP header.

25 SoapHeaderAttribute

*Example Syntax:*

ToString

```
[C#] public SoapHeaderAttribute(string memberName);  
[C++] public: SoapHeaderAttribute(String* memberName);  
[VB] Public Sub New(ByVal memberName As String)  
[JScript] public function SoapHeaderAttribute(memberName : String);
```

*Description*

Initializes a new instance of the **System.Web.Services.Protocols.SoapHeaderAttribute** class setting the member of the Web Service class representing the SOAP header contents.

A Web Service client or Web Service can specify the SOAP headers it wants to process for specific Web Service methods by applying a **System.Web.Services.Protocols.SoapHeaderAttribute** to the Web Service method in the Web Service class or the corresponding method in the proxy class. In order for the Web Service method to receive the contents of the SOAP header, a member is added to the Web Service class of a **System.Type** derived from **System.Web.Services.Protocols.SoapHeader** . Likewise, a Web Service client adds a member to the proxy class deriving from **System.Web.Services.Protocols.SoapHeader** . That **System.Type** deriving from **System.Web.Services.Protocols.SoapHeader** represents the contents of the SOAP header and is created by the developer of the Web Service or Web Service client that wants to receive the SOAP header. Once the class is created, a **System.Web.Services.Protocols.SoapHeaderAttribute** can be applied to the

1 Web Service method or corresponding method in the Web Service client proxy  
2 class specifying the member to receive the SOAP header contents with the  
3 *memberName* parameter to this constructor or the

4 **System.Web.Services.Protocols.SoapHeaderAttribute.MemberName** property.

5 Member of the Web Service class representing the SOAP header contents. Sets the

6 **System.Web.Services.Protocols.SoapHeaderAttribute.MemberName** property.

7 Direction

8 ToString

9  
10 [C#] public SoapHeaderDirection Direction {get; set;}

11 [C++] public: \_\_property SoapHeaderDirection get\_Direction();public: \_\_property  
12 void set\_Direction(SoapHeaderDirection);

13 [VB] Public Property Direction As SoapHeaderDirection

14 [JScript] public function get Direction() : SoapHeaderDirection;public function set  
15 Direction(SoapHeaderDirection);

16  
17 *Description*

18 Gets or sets whether the SOAP header is intended for the Web Service or  
19 the Web Service client or both.

20 MemberName

21 ToString

22  
23 [C#] public string MemberName {get; set;}

24 [C++] public: \_\_property String\* get\_MemberName();public: \_\_property void  
25 set\_MemberName(String\*);

1 [VB] Public Property MemberName As String

2 [JScript] public function get MemberName() : String;public function set

3 MemberName(String);

4  
5 *Description*

6 Gets or sets the member of the Web Service class representing the SOAP  
7 header contents.

8 A Web Service client or Web Service can specify the SOAP headers it  
9 wants to process for specific Web Service methods by applying a

10 **System.Web.Services.Protocols.SoapHeaderAttribute** to the Web Service  
11 method in the Web Service class or the corresponding method in the proxy class.

12 In order for the Web Service method to receive the contents of the SOAP header, a  
13 member is added to the Web Service class of a **System.Type** derived from

14 **System.Web.Services.Protocols.SoapHeader** . Likewise, a Web Service client  
15 adds a member to the proxy class deriving from

16 **System.Web.Services.Protocols.SoapHeader** . That **System.Type** deriving from  
17 **System.Web.Services.Protocols.SoapHeader** represents the contents of the

18 SOAP header and is created by the developer of the Web Service or Web Service  
19 client that wants to receive the SOAP header. Once the class is created, a

20 **System.Web.Services.Protocols.SoapHeaderAttribute** can be applied to the  
21 Web Service method or corresponding method in the Web Service client proxy

22 class specifying the member to receive the SOAP header contents with the

23 **System.Web.Services.Protocols.SoapHeaderAttribute.MemberName** property.

24 Required

25 ToString

1  
2 [C#] public bool Required {get; set;}

3 [C++] public: \_\_property bool get\_Required();public: \_\_property void  
4 set\_Required(bool);

5 [VB] Public Property Required As Boolean

6 [JScript] public function get Required() : Boolean;public function set  
7 Required(Boolean);

8  
9 *Description*

10 Gets or sets a value indicating whether the SOAP header must be  
11 understood and processed by the recipient Web Service or Web Service client.

12 When the **System.Type** of the member (represented by the  
13 **System.Web.Services.Protocols.SoapHeaderAttribute.MemberName** property)  
14 is either **System.Web.Services.Protocols.SoapHeader** or  
15 **System.Web.Services.Protocols.SoapUnknownHeader** , the  
16 **System.Web.Services.Protocols.SoapHeaderAttribute.Required** property must  
17 be **false** . For types that derive from  
18 **System.Web.Services.Protocols.SoapHeader** , it is fine for the  
19 **System.Web.Services.Protocols.SoapHeaderAttribute.Required** property to be  
20 **true** or **false** .

21 TypeId

22 SoapHeaderCollection class (System.Web.Services.Protocols)

23 ToString

1  
2  
3 *Description*

4       Contains a collection of **System.Web.Services.Protocols.SoapHeader**  
5 objects.

6       SoapHeaderCollection

7       *Example Syntax:*

8       ToString

9  
10 [C#] public SoapHeaderCollection();

11 [C++] public: SoapHeaderCollection();

12 [VB] Public Sub New()

13 [JScript] public function SoapHeaderCollection();

14       Count

15       InnerList

16       Item

17       ToString

18  
19  
20 *Description*

21       Gets or sets the **System.Web.Services.Protocols.SoapHeader** at position  
22 *index* of the **System.Web.Services.Protocols.SoapHeaderCollection** .

23       This property provides the ability to access a specific element in the  
24 collection by using the following syntax: myCollection[index] . The zero-based  
25 index of the **System.Web.Services.Protocols.SoapHeader** to get or set.

List

Add

[C#] public int Add(SoapHeader header);

[C++] public: int Add(SoapHeader\* header);

[VB] Public Function Add(ByVal header As SoapHeader) As Integer

[JScript] public function Add(header : SoapHeader) : int;

*Description*

Adds a **System.Web.Services.Protocols.SoapHeader** to the **System.Web.Services.Protocols.SoapHeaderCollection**.

*Return Value:* The position into which the **System.Web.Services.Protocols.SoapHeader** was inserted. The **System.Web.Services.Protocols.SoapHeader** to add to the **System.Web.Services.Protocols.SoapHeaderCollection**.

Contains

[C#] public bool Contains(SoapHeader header);

[C++] public: bool Contains(SoapHeader\* header);

[VB] Public Function Contains(ByVal header As SoapHeader) As Boolean

[JScript] public function Contains(header : SoapHeader) : Boolean;

*Description*

Determines whether the **System.Web.Services.Protocols.SoapHeaderCollection** contains a specific

## **System.Web.Services.Protocols.SoapHeader** ..

*Return Value:* **true** if the **System.Web.Services.Protocols.SoapHeader** is found in the **System.Web.Services.Protocols.SoapHeaderCollection** ; otherwise, **false** . The **System.Web.Services.Protocols.SoapHeader** to locate in the **System.Web.Services.Protocols.SoapHeaderCollection**.

### **CopyTo**

[C#] public void CopyTo(SoapHeader[] array, int index);

[C++] public: void CopyTo(SoapHeader\* array[], int index);

[VB] Public Sub CopyTo(ByVal array() As SoapHeader, ByVal index As Integer)

[JScript] public function CopyTo(array : SoapHeader[], index : int);

### *Description*

Copies the elements of the **System.Web.Services.Protocols.SoapHeaderCollection** to an **System.Array** , starting at a particular **System.Array** index. The one-dimensional **System.Array** that is the destination of the elements copied from **System.Web.Services.Protocols.SoapHeaderCollection**. The **System.Array** must have zero-based indexing. The zero-based index in *array* at which copying begins.

### **IndexOf**

[C#] public int IndexOf(SoapHeader header);

[C++] public: int IndexOf(SoapHeader\* header);

[VB] Public Function IndexOf(ByVal header As SoapHeader) As Integer



1 [JScript] public function IndexOf(header : SoapHeader) : int;

3 *Description*

4 Determines the index of the SoapHeader in the SoapHeaderCollection.

5 *Return Value:* The index of *header* if found in the

6 **System.Web.Services.Protocols.SoapHeaderCollection** ; otherwise, -1. The

7 **System.Web.Services.Protocols.SoapHeader** to locate in the

8 **System.Web.Services.Protocols.SoapHeaderCollection**.

9 Insert

11 [C#] public void Insert(int index, SoapHeader header);

12 [C++] public: void Insert(int index, SoapHeader\* header);

13 [VB] Public Sub Insert(ByVal index As Integer, ByVal header As SoapHeader)

14 [JScript] public function Insert(index : int, header : SoapHeader);

16 *Description*

17 Inserts a **System.Web.Services.Protocols.SoapHeader** into the

18 **System.Web.Services.Protocols.SoapHeaderCollection** at the specified index.

19 If *index* equals the number of items in the collection, then the

20 **System.Web.Services.Protocols.SoapHeader** is appended to the end. The zero-

21 based index at which the **System.Web.Services.Protocols.SoapHeader** is

22 inserted into the **System.Web.Services.Protocols.SoapHeaderCollection** . The

23 **System.Web.Services.Protocols.SoapHeader** inserted into the

24 **System.Web.Services.Protocols.SoapHeaderCollection** .

25 Remove

```

1 [C#] public void Remove(SoapHeader header);
2
3 [C++] public: void Remove(SoapHeader* header);
4
5 [VB] Public Sub Remove(ByVal header As SoapHeader)
6
7 [JScript] public function Remove(header : SoapHeader);
8

```

### *Description*

Removes the first occurrence of a specific **System.Web.Services.Protocols.SoapHeader** from the **System.Web.Services.Protocols.SoapHeaderCollection**.

The **System.Web.Services.Protocols.SoapHeader** elements that follow the removed **System.Web.Services.Protocols.SoapHeader** move up to occupy the vacated spot. The **System.Web.Services.Protocols.SoapHeader** to remove from the **System.Web.Services.Protocols.SoapHeaderCollection**.

SoapHeaderDirection enumeration (System.Web.Services.Protocols)

ToString

### *Description*

Specifies whether the recipient of the **System.Web.Services.Protocols.SoapHeader** is the Web Service, the Web Service client or both.

A **System.Web.Services.Protocols.SoapHeaderAttribute** can be applied to either a Web Service method or a method on a proxy class to a Web Service. In either case, the recipient is specified by the

1 **System.Web.Services.Protocols.SoapHeaderAttribute.Direction** property of

2 **System.Web.Services.Protocols.SoapHeaderAttribute** .

3 ToString

4  
5 [C#] public const SoapHeaderDirection In;

6 [C++] public: const SoapHeaderDirection In;

7 [VB] Public Const In As SoapHeaderDirection

8 [JScript] public var In : SoapHeaderDirection;

9  
10 *Description*

11 Specifies the **System.Web.Services.Protocols.SoapHeader** is sent to the  
12 Web Service.

13 ToString

14  
15 [C#] public const SoapHeaderDirection InOut;

16 [C++] public: const SoapHeaderDirection InOut;

17 [VB] Public Const InOut As SoapHeaderDirection

18 [JScript] public var InOut : SoapHeaderDirection;

19  
20 *Description*

21 Specifies the **System.Web.Services.Protocols.SoapHeader** is sent to both  
22 the Web Service and the Web Service client.

23 ToString

24  
25 [C#] public const SoapHeaderDirection Out;

```

1 [C++] public: const SoapHeaderDirection Out;
2 [VB] Public Const Out As SoapHeaderDirection
3 [JScript] public var Out : SoapHeaderDirection;

```

#### *Description*

Specifies the **System.Web.Services.Protocols.SoapHeader** is sent to the Web Service client.

**SoapHeaderException** class (System.Web.Services.Protocols)

**ToString**

#### *Description*

The exception that is thrown when a Web Service method is called over Simple Object Access Protocol (SOAP) and an exception occurs processing the SOAP header.

When a Web Service client adds a SOAP header to a Web Service method call with the **System.Web.Services.Protocols.SoapHeader.MustUnderstand** property set to **true** , the Web Service method must set the **System.Web.Services.Protocols.SoapHeader.DidUnderstand** property to **true**; otherwise a **System.Web.Services.Protocols.SoapHeaderException** is thrown.

**SoapHeaderException**

*Example Syntax:*

**ToString**

```

1
2 [C#] public SoapHeaderException(string message, XmlQualifiedName code);
3 [C++] public: SoapHeaderException(String* message, XmlQualifiedName*
4 code);
5 [VB] Public Sub New(ByVal message As String, ByVal code As
6 XmlQualifiedName)
7 [JScript] public function SoapHeaderException(message : String, code :
8 XmlQualifiedName);
9

```

### *Description*

Initializes a new instance of the **System.Web.Services.Protocols.SoapHeaderException** class. A message that identifies the reason the exception occurred. Sets the **System.Exception.Message** property. The type of error that occurred. Sets the **System.Web.Services.Protocols.SoapException.Code** property.

**SoapHeaderException**

### *Example Syntax:*

**ToString**

```

20 [C#] public SoapHeaderException(string message, XmlQualifiedName code,
21 Exception innerException);
22 [C++] public: SoapHeaderException(String* message, XmlQualifiedName* code,
23 Exception* innerException);
24 [VB] Public Sub New(ByVal message As String, ByVal code As
25 XmlQualifiedName, ByVal innerException As Exception)

```

1 [JScript] public function SoapHeaderException(message : String, code :  
2 XmlQualifiedName, innerException : Exception);

3  
4 *Description*

5       Initializes a new instance of the  
6 **System.Web.Services.Protocols.SoapHeaderException** class. A message that  
7 identifies the reason the exception occurred. Sets the **System.Exception.Message**  
8 property. The type of error that occurred. Sets the  
9 **System.Web.Services.Protocols.SoapException.Code** property. A reference to  
10 the root cause of an exception. Sets the **System.Exception.InnerException**  
11 property.

12       SoapHeaderException

13       *Example Syntax:*

14       ToString

15  
16 [C#] public SoapHeaderException(string message, XmlQualifiedName code,  
17 string actor);

18 [C++] public: SoapHeaderException(String\* message, XmlQualifiedName\* code,  
19 String\* actor);

20 [VB] Public Sub New(ByVal message As String, ByVal code As  
21 XmlQualifiedName, ByVal actor As String)

22 [JScript] public function SoapHeaderException(message : String, code :  
23 XmlQualifiedName, actor : String);

24  
25 *Description*

1        Initializes a new instance of the  
2        **System.Web.Services.Protocols.SoapHeaderException** class. A message that  
3        identifies the reason the exception occurred. Sets the **System.Exception.Message**  
4        property. The type of error that occurred. Sets the  
5        **System.Web.Services.Protocols.SoapException.Code** property. The piece of  
6        code that caused the exception. Typically, a URL to a Web Service method. Sets  
7        the **System.Web.Services.Protocols.SoapException.Actor** property.

8        SoapHeaderException

9        *Example Syntax:*

10       ToString

11  
12       [C#] public SoapHeaderException(string message, XmlQualifiedName code,  
13       string actor, Exception innerException);  
14       [C++] public: SoapHeaderException(String\* message, XmlQualifiedName\* code,  
15       String\* actor, Exception\* innerException);  
16       [VB] Public Sub New(ByVal message As String, ByVal code As  
17       XmlQualifiedName, ByVal actor As String, ByVal innerException As Exception)  
18       [JScript] public function SoapHeaderException(message : String, code :  
19       XmlQualifiedName, actor : String, innerException : Exception);

20  
21       *Description*

22       Initializes a new instance of the  
23       **System.Web.Services.Protocols.SoapHeaderException** class. A message that  
24       identifies the reason the exception occurred. Sets the **System.Exception.Message**  
25       property. The type of error that occurred. Sets the

**System.Web.Services.Protocols.SoapException.Code** property. The piece of code that caused the exception. Typically, a URL to a Web Service method. Sets the **System.Web.Services.Protocols.SoapException.Actor** property. A reference to the root cause of an exception. Sets the **System.Exception.InnerException** property.

**SoapHeaderException**

*Example Syntax:*

**ToString**

```
[C#] public SoapHeaderException(string message, XmlQualifiedName code,
string actor, XmlNode[] otherElements);
[C++] public: SoapHeaderException(String* message, XmlQualifiedName* code,
String* actor, XmlNode* otherElements[]);
[VB] Public Sub New(ByVal message As String, ByVal code As
XmlQualifiedName, ByVal actor As String, ByVal otherElements() As XmlNode)
[JScript] public function SoapHeaderException(message : String, code :
XmlQualifiedName, actor : String, otherElements : XmlNode[]);
```

### *Description*

Initializes a new instance of the

**System.Web.Services.Protocols.SoapHeaderException** class. A message that identifies the reason the exception occurred. Sets the **System.Exception.Message** property. The type of error that occurred. Sets the **System.Web.Services.Protocols.SoapException.Code** property. The piece of code that caused the exception. Typically, a URL to a Web Service method. Sets



1 the **System.Web.Services.Protocols.SoapException.Actor** property. Optional  
2 exception information. Sets the  
3 **System.Web.Services.Protocols.SoapException.OtherElements** property.

4 SoapHeaderException

5 *Example Syntax:*

6 ToString

7  
8 [C#] public SoapHeaderException(string message, XmlQualifiedName code,  
9 string actor, XmlNode[] otherElements, Exception innerException);

10 [C++] public: SoapHeaderException(String\* message, XmlQualifiedName\* code,  
11 String\* actor, XmlNode\* otherElements[], Exception\* innerException);

12 [VB] Public Sub New(ByVal message As String, ByVal code As  
13 XmlQualifiedName, ByVal actor As String, ByVal otherElements() As XmlNode,  
14 ByVal innerException As Exception)

15 [JScript] public function SoapHeaderException(message : String, code :  
16 XmlQualifiedName, actor : String, otherElements : XmlNode[], innerException :  
17 Exception);

18  
19 *Description*

20 Initializes a new instance of the  
21 **System.Web.Services.Protocols.SoapHeaderException** class. A message that  
22 identifies the reason the exception occurred. Sets the **System.Exception.Message**  
23 property. The type of error that occurred. Sets the  
24 **System.Web.Services.Protocols.SoapException.Code** property. The piece of  
25 code that caused the exception. Typically, a URL to a Web Service method. Sets

1 the **System.Web.Services.Protocols.SoapException.Actor** property. Optional  
2 exception information. Sets the  
3 **System.Web.Services.Protocols.SoapException.OtherElements** property. A  
4 reference to the root cause of an exception. Sets the  
5 **System.Exception.InnerException** property.

6 Actor

7 Code

8 Detail

9 HelpLink

10 HResult

11 InnerException

12 Message

13 OtherElements

14 Source

15 StackTrace

16 TargetSite

17 SoapHttpClientProtocol class (System.Web.Services.Protocols)

18 ToString

21 *Description*

22 Specifies the class ASP.NET Web Service client proxies derive from when  
23 using Simple Object Access Protocol (SOAP).

24 ASP.NET incorporates two distinct Web Services functionalities: Building  
25 ASP.NET Web Services and Building Web Services clients. If you are building a

Web Service client using ASP.NET, then a proxy class deriving indirectly or directly from **System.Web.Services.Protocols.WebClientProtocol** needs to be created for the Web Service you want to call. When the Web Service client is calling via SOAP, the proxy class should derive from **System.Web.Services.Protocols.SoapHttpClientProtocol**, which derives from **System.Web.Services.Protocols.HttpWebClientProtocol**.

SoapHttpClientProtocol

*Example Syntax:*

ToString

[C#] public SoapHttpClientProtocol();

[C++] public: SoapHttpClientProtocol();

[VB] Public Sub New()

[JScript] public function SoapHttpClientProtocol();

*Description*

Initializes a new instance of the **System.Web.Services.Protocols.SoapHttpClientProtocol** class.

AllowAutoRedirect

ClientCertificates

ConnectionGroupName

Container

CookieContainer

Credentials

DesignMode

1	Events
2	PreAuthenticate
3	Proxy
4	RequestEncoding
5	Site
6	Timeout
7	Url
8	UserAgent
9	BeginInvoke
10	
11	[C#] protected IAsyncResult BeginInvoke(string methodName, object[]
12	parameters, AsyncCallback callback, object asyncState);
13	[C++] protected: IAsyncResult* BeginInvoke(String* methodName, Object*
14	parameters __gc[], AsyncCallback* callback, Object* asyncState);
15	[VB] Protected Function BeginInvoke(ByVal methodName As String, ByVal
16	parameters() As Object, ByVal callback As AsyncCallback, ByVal asyncState As
17	Object) As IAsyncResult
18	[JScript] protected function BeginInvoke(methodName : String, parameters :
19	Object[], callback : AsyncCallback, asyncState : Object) : IAsyncResult;
20	
21	<i>Description</i>
22	Starts an asynchronous invocation of a Web Service method using Simple
23	Object Access Protocol(SOAP).
24	<i>Return Value:</i> An <b>System.IAsyncResult</b> which is passed to
25	

1 **System.Web.Services.Protocols.SoapHttpClientProtocol.EndInvoke(System.I**  
2 **AsyncResult)** to obtain the return values from the remote method call.

3 Typically, you would not call the  
4 **System.Web.Services.Protocols.SoapHttpClientProtocol.BeginInvoke(System.**  
5 **String,System.Object[],System.AsyncCallback,System.Object)** method  
6 directly, unless you were building your own proxy class for a Web Service. The  
7 name of the Web Service method in the derived class that is invoking  
8 **System.Web.Services.Protocols.SoapHttpClientProtocol.BeginInvoke(System.**  
9 **String,System.Object[],System.AsyncCallback,System.Object)**. An array of  
10 objects containing the parameters to pass to the remote web service. The order of  
11 the values in the array correspond to the order of the parameters in the calling  
12 method of the derived class. The delegate to call when the asynchronous invoke is  
13 complete. If *callback* is **null**, the delegate is not called. Extra information supplied  
14 by the caller.

15 Discover

16  
17 [C#] public void Discover();  
18 [C++] public: void Discover();  
19 [VB] Public Sub Discover()  
20 [JScript] public function Discover();  
21

## 22 *Description*

23 Dynamically binds to a Web Service described in the discovery document  
24 at **System.Web.Services.Protocols.WebClientProtocol.Url** .  
25

1 A Web Service client can dynamically bind to a Web Service other than the  
2 one referenced in the proxy class using the

3 **System.Web.Services.Protocols.SoapHttpClientProtocol.Discover** method.

4 Typically, the **System.Web.Services.Protocols.WebClientProtocol.Url** property  
5 refers to the base address of the Web Service, however prior to invoking the  
6 **System.Web.Services.Protocols.SoapHttpClientProtocol.Discover** method, set  
7 the **System.Web.Services.Protocols.WebClientProtocol.Url** property to a URL  
8 of a discovery document. Invoking the

9 **System.Web.Services.Protocols.SoapHttpClientProtocol.Discover** method  
10 attempts to find a match in the discovery document to the binding defined in the  
11 proxy class and then dynamically bind to it. If successful, future method  
12 invocations are directed to the Web Service described in the discovery document.

13 **EndInvoke**

14  
15 [C#] protected object[] EndInvoke(IAsyncResult asyncResult);

16 [C++] protected: Object\* EndInvoke(IAsyncResult\* asyncResult) \_\_gc[];

17 [VB] Protected Function EndInvoke(ByVal asyncResult As IAsyncResult) As  
18 Object()

19 [JScript] protected function EndInvoke(asyncResult : IAsyncResult) : Object[];

20  
21 *Description*

22 Ends an asynchronous invocation of a Web Service method using Simple  
23 Object Access Protocol(SOAP).

24 *Return Value:* An array of objects containing the return value and any by-  
25 reference or out parameters of the derived class method.

Typically, you would not call the **System.Web.Services.Protocols.SoapHttpClientProtocol.EndInvoke(System.IAsyncResult)** method directly, unless you were building your own proxy class for a Web Service. The **System.IAsyncResult** returned from **System.Web.Services.Protocols.SoapHttpClientProtocol.BeginInvoke(System.String,System.Object[],System.AsyncCallback,System.Object)**.

### GetWebRequest

[C#] protected override WebRequest GetWebRequest(Uri uri);

[C++] protected: WebRequest\* GetWebRequest(Uri\* uri);

[VB] Overrides Protected Function GetWebRequest(ByVal uri As Uri) As

WebRequest

[JScript] protected override function GetWebRequest(uri : Uri) : WebRequest;

### Description

Creates a **System.Net.WebRequest** instance for the specified *url* .

**Return Value:** The **System.Net.WebRequest** instance.

This method can be overridden to customize the **System.Net.WebRequest** object before the Web Service request is made. For example you could add a custom header to the request. The **System.Uri** to use when creating the **System.Net.WebRequest**.

### Invoke

[C#] protected object[] Invoke(string methodName, object[] parameters);

[C++] protected: Object\* Invoke(String\* methodName, Object\* parameters

1 \_\_gc[]) \_\_gc[];

2 [VB] Protected Function Invoke(ByVal methodName As String, ByVal

3 parameters() As Object) As Object()

4 [JScript] protected function Invoke(methodName : String, parameters : Object[]) :

5 Object[];

6  
7 *Description*

8 Invokes a Web Service method synchronously using SOAP.

9 *Return Value:* An array of objects containing the return value and any by reference  
10 or out parameters of the derived class method.

11 Typically, you would not call the

12 **System.Web.Services.Protocols.SoapHttpClientProtocol.Invoke(System.Strin**  
13 **g,System.Object[])** method directly, unless you were building your own proxy  
14 class for a Web Service. The name of the Web Service method. An array of  
15 objects containing the parameters to pass to the remote web service. The order of  
16 the values in the array correspond to the order of the parameters in the calling  
17 method of the derived class.

18 SoapMessage class (System.Web.Services.Protocols)

19 ToString

20  
21  
22 *Description*

23 Represents the data in a SOAP request or SOAP response at a specific

24 **System.Web.Services.Protocols.SoapMessageStage** .



The **System.Web.Services.Protocols.SoapMessage** class is primary used for SOAP extensions to represent the data in a SOAP request or SOAP response at the **System.Web.Services.Protocols.SoapMessageStage** in the **System.Web.Services.Protocols.SoapMessage.Stage** property. A SOAP extension, which is a class deriving from **System.Web.Services.Protocols.SoapExtension** , receives a **System.Web.Services.Protocols.SoapMessage** at each **System.Web.Services.Protocols.SoapMessageStage** when the **System.Web.Services.Protocols.SoapExtension.ProcessMessage(System.Web.Services.Protocols.SoapMessage)** method is called. It is up to the particular SOAP extension to decide how to process the **System.Web.Services.Protocols.SoapMessage** , but a couple of common SOAP extensions might include encryption and compression.

Action

ToString

[C#] public abstract string Action {get;}

[C++] public: \_\_property virtual String\* get\_Action() = 0;

[VB] MustOverride Public ReadOnly Property Action As String

[JScript] public abstract function get Action() : String;

### *Description*

When overridden in a derived class, gets the SOAPAction HTTP request header field for the SOAP request or SOAP response.

1 The **System.Web.Services.Protocols.SoapMessage.Action** property can  
2 be accessed during any **System.Web.Services.Protocols.SoapMessageStage** .

3 **ContentType**

4 **ToString**

5  
6 [C#] public string ContentType {get; set;}

7 [C++] public: \_\_property String\* get\_ContentType();public: \_\_property void

8 set\_ContentType(String\*);

9 [VB] Public Property ContentType As String

10 [JScript] public function get ContentType() : String;public function set

11 ContentType(String);

12  
13 *Description*

14 Gets or sets the HTTP Content-Type of the SOAP request or SOAP  
15 response.

16 The **System.Web.Services.Protocols.SoapMessage.ContentType**  
17 property can only be accessed in the  
18 **System.Web.Services.Protocols.SoapMessageStage.BeforeSerialize** and  
19 **System.Web.Services.Protocols.SoapMessageStage.BeforeDeserialize** stages;  
20 otherwise an **System.InvalidOperationException** is thrown.

21 **Exception**

22 **ToString**

23  
24 [C#] public SoapException Exception {get;}

25 [C++] public: \_\_property SoapException\* get\_Exception();

1 [VB] Public ReadOnly Property Exception As SoapException

2 [JScript] public function get Exception() : SoapException;

3  
4 *Description*

5 Gets the **System.Web.Services.Protocols.SoapException** from the call to  
6 the Web Service method.

7 The **System.Web.Services.Protocols.SoapMessage.Exception** property  
8 can be accessed during any **System.Web.Services.Protocols.SoapMessageStage**

9  
10 Headers

11 ToString

12  
13 [C#] public SoapHeaderCollection Headers {get;}

14 [C++] public: \_\_property SoapHeaderCollection\* get\_Headers();

15 [VB] Public ReadOnly Property Headers As SoapHeaderCollection

16 [JScript] public function get Headers() : SoapHeaderCollection;

17  
18 *Description*

19 A collection of the SOAP headers applied to the current SOAP request or  
20 SOAP response.

21 SOAP headers are an optional portion of a SOAP request or SOAP  
22 response. For more details on using SOAP headers in ASP.NET Web Services and  
23 ASP.NET clients, see the **System.Web.Services.Protocols.SoapHeader** class.

24 MethodInfo

25 ToString

```

1 [C#] public abstract LogicalMethodInfo MethodInfo {get;}
2 [C++] public: __property virtual LogicalMethodInfo* get_MethodInfo() = 0;
3 [VB] MustOverride Public ReadOnly Property MethodInfo As LogicalMethodInfo
4 [JScript] public abstract function get MethodInfo() : LogicalMethodInfo;
5
6

```

### *Description*

When overridden in a derived class, gets a representation of the method prototype for the Web Service method for which the SOAP request is intended.

Although the **System.Web.Services.Protocols.SoapMessage.MethodInfo** can be accessed during any **System.Web.Services.Protocols.SoapMessageStage**, it only has data during

**System.Web.Services.Protocols.SoapMessageStage.AfterDeserialize** and **System.Web.Services.Protocols.SoapMessageStage.BeforeSerialize**.

OneWay

ToString

```

18 [C#] public abstract bool OneWay {get;}
19 [C++] public: __property virtual bool get_OneWay() = 0;
20 [VB] MustOverride Public ReadOnly Property OneWay As Boolean
21 [JScript] public abstract function get OneWay() : Boolean;
22

```

### *Description*

Gets a value indicating the **System.Web.Services.Protocols.SoapDocumentMethodAttribute.OneWay**

property of either the

**System.Web.Services.Protocols.SoapDocumentMethodAttribute** or

**System.Web.Services.Protocols.SoapRpcMethodAttribute** applied to the Web Service method.

See the

**System.Web.Services.Protocols.SoapDocumentMethodAttribute.OneWay**

property of **System.Web.Services.Protocols.SoapDocumentMethodAttribute** or

**System.Web.Services.Protocols.SoapRpcMethodAttribute** for details about one-way Web Service methods.

Stage

ToString

[C#] public SoapMessageStage Stage {get;}

[C++] public: \_\_property SoapMessageStage get\_Stage();

[VB] Public ReadOnly Property Stage As SoapMessageStage

[JScript] public function get Stage() : SoapMessageStage;

### *Description*

Gets the **System.Web.Services.Protocols.SoapMessageStage** of the **System.Web.Services.Protocols.SoapMessage** .

The **System.Web.Services.Protocols.SoapMessage.Stage** property can be accessed at any **System.Web.Services.Protocols.SoapMessageStage** .

Stream

ToString

```

1
2 [C#] public Stream Stream {get;}
3 [C++] public: __property Stream* get_Stream();
4 [VB] Public ReadOnly Property Stream As Stream
5 [JScript] public function get Stream() : Stream;
6

```

### 7 *Description*

8 Gets the data representing the SOAP request or SOAP response in the form  
9 of a **System.IO.Stream** object.

10 SOAP extensions that inspect the data representing a SOAP request or  
11 SOAP response, such as a logging SOAP extension, can use the  
12 **System.IO.Stream** property to view the data at each  
13 **System.Web.Services.Protocols.SoapMessageStage** .

14 Url

15 ToString

```

16
17 [C#] public abstract string Url {get;}
18 [C++] public: __property virtual String* get_Url() = 0;
19 [VB] MustOverride Public ReadOnly Property Url As String
20 [JScript] public abstract function get Url() : String;
21

```

### 22 *Description*

23 When overridden in a derived class, gets the base URL of the Web Service.

24 The **System.Web.Services.Protocols.SoapMessage.Url** property can be  
25 accessed during any **System.Web.Services.Protocols.SoapMessageStage** .

## EnsureInStage

[C#] protected abstract void EnsureInStage();

[C++] protected: virtual void EnsureInStage() = 0;

[VB] MustOverride Protected Sub EnsureInStage()

[JScript] protected abstract function EnsureInStage();

### *Description*

When overridden in a derived class, asserts that the current **System.Web.Services.Protocols.SoapMessageStage** stage is a stage where in parameters are available.

The asserted stage is different for Web Service clients and Web Service methods. A Web Service method's in parameters are available after the SOAP request is deserialized in the **System.Web.Services.Protocols.SoapMessageStage.AfterDeserialize** stage. For a Web Service client, the in parameters are available prior to the SOAP request serialization process in the **System.Web.Services.Protocols.SoapMessageStage.BeforeSerialize** stage.

## EnsureOutStage

[C#] protected abstract void EnsureOutStage();

[C++] protected: virtual void EnsureOutStage() = 0;

[VB] MustOverride Protected Sub EnsureOutStage()

[JScript] protected abstract function EnsureOutStage();

## *Description*

When overridden in a derived class, asserts that the current **System.Web.Services.Protocols.SoapMessageStage** stage is a stage where out parameters are available.

The asserted stage is different for Web Service clients and Web Service methods. A Web Service method's out parameters are available prior to the SOAP response serialization process in the

**System.Web.Services.Protocols.SoapMessageStage.BeforeSerialize** stage. For a Web Service client, the out parameters are available after the SOAP response deserialization process in the

**System.Web.Services.Protocols.SoapMessageStage.AfterDeserialize** stage.

## *EnsureStage*

[C#] protected void EnsureStage(SoapMessageStage stage);

[C++] protected: void EnsureStage(SoapMessageStage stage);

[VB] Protected Sub EnsureStage(ByVal stage As SoapMessageStage)

[JScript] protected function EnsureStage(stage : SoapMessageStage);

## *Description*

Ensures that the **System.Web.Services.Protocols.SoapMessageStage** of the call to the Web Service method is the stage or stages passed in. If the current processing stage is not one of the stages passed in, an exception is thrown.



Multiple stages can be asserted by performing a bitwise OR operation on multiple **System.Web.Services.Protocols.SoapMessageStage** stages. The **System.Web.Services.Protocols.SoapMessageStage** asserted.

#### GetInParameterValue

[C#] public object GetInParameterValue(int index);  
[C++] public: Object\* GetInParameterValue(int index);  
[VB] Public Function GetInParameterValue(ByVal index As Integer) As Object  
[JScript] public function GetInParameterValue(index : int) : Object;

#### *Description*

Gets the parameter passed into the Web Service method at the specified index.

*Return Value:* An **System.Object** representing the parameter at index in the array of parameters.

The **System.Web.Services.Protocols.SoapMessage.GetInParameterValue(System.Int32)** property is only accessible when in parameters are available. ASP.NET Web Services throws **System.InvalidOperationException** when accessing the **System.Web.Services.Protocols.SoapMessage.GetInParameterValue(System.Int32)** property if in parameters are not available. The zero-based index of the parameter in the array of parameters .

#### GetOutParameterValue

[C#] public object GetOutParameterValue(int index);

1 [C++] public: Object\* GetOutParameterValue(int index);

2 [VB] Public Function GetOutParameterValue(ByVal index As Integer) As Object

3 [JScript] public function GetOutParameterValue(index : int) : Object;

4  
5 *Description*

6 Gets the out parameter passed into the Web Service method at the specified  
7 index.

8 *Return Value:* An **System.Object** representing the parameter at index in the array  
9 of parameters.

10 The  
11 **System.Web.Services.Protocols.SoapMessage.GetInParameterValue(System.I**  
12 **nt32)** property is only accessible when out parameters are available. ASP.NET  
13 Web Services throws **System.InvalidOperationException** when accessing the  
14 **System.Web.Services.Protocols.SoapMessage.GetOutParameterValue(System**  
15 **.Int32)** property if out parameters are not available. The zero-based index of the  
16 parameter in the array of parameters.

17 **GetReturnValue**

18  
19 [C#] public object GetReturnValue();

20 [C++] public: Object\* GetReturnValue();

21 [VB] Public Function GetReturnValue() As Object

22 [JScript] public function GetReturnValue() : Object;

23  
24 *Description*

25

1 Gets the return value of a Web Service method.  
2 *Return Value:* An **System.Object** representing the return value of the Web Service  
3 method.

4 The **System.Web.Services.Protocols.SoapMessage.GetReturnValue**  
5 property is only accessible when a return value is available. ASP.NET Web  
6 Services throws **System.InvalidOperationException** when accessing the  
7 **System.Web.Services.Protocols.SoapMessage.GetReturnValue** property and  
8 the return value is not available.

9 SoapMessageStage enumeration (System.Web.Services.Protocols)

10 ToString

11  
12  
13 *Description*

14 Specifies the processing stage of a SOAP message.

15 ASP.NET Web Services provides an extensibility mechanism for calling  
16 Web Services using SOAP. The extensibility mechanism revolves around a  
17 **System.Web.Services.Protocols.SoapExtension** that is allowed to inspect or  
18 modify a message at specific stages in message processing on either the client or  
19 the server. This enum specifies the processing stage of a  
20 **System.Web.Services.Protocols.SoapMessage** .

21 ToString

22  
23 [C#] public const SoapMessageStage AfterDeserialize;

24 [C++] public: const SoapMessageStage AfterDeserialize;

25 [VB] Public Const AfterDeserialize As SoapMessageStage

1 [JScript] public var AfterDeserialize : SoapMessageStage;

3 *Description*

4 The stage just after a **System.Web.Services.Protocols.SoapMessage** is  
5 deserialized from a SOAP message into a object.

6 ToString

8 [C#] public const SoapMessageStage AfterSerialize;

9 [C++] public: const SoapMessageStage AfterSerialize;

10 [VB] Public Const AfterSerialize As SoapMessageStage

11 [JScript] public var AfterSerialize : SoapMessageStage;

13 *Description*

14 The stage just after a **System.Web.Services.Protocols.SoapMessage** is  
15 serialized, but before the SOAP message is sent over the wire.

16 ToString

18 [C#] public const SoapMessageStage BeforeDeserialize;

19 [C++] public: const SoapMessageStage BeforeDeserialize;

20 [VB] Public Const BeforeDeserialize As SoapMessageStage

21 [JScript] public var BeforeDeserialize : SoapMessageStage;

23 *Description*

24 The stage just before a **System.Web.Services.Protocols.SoapMessage** is  
25 deserialized from the SOAP message sent across the network into a object.

1 ToString

2

3 [C#] public const SoapMessageStage BeforeSerialize;

4 [C++] public: const SoapMessageStage BeforeSerialize;

5 [VB] Public Const BeforeSerialize As SoapMessageStage

6 [JScript] public var BeforeSerialize : SoapMessageStage;

7

8 *Description*

9 The stage just prior to a **System.Web.Services.Protocols.SoapMessage**

10 being serialized.

11 SoapParameterStyle enumeration (System.Web.Services.Protocols)

12 ToString

13

14

15 *Description*

16 Specifies how parameters are encoded in the XML portion of a SOAP

17 message.

18 **System.Web.Services.Protocols.SoapParameterStyle** can be set when

19 applying a **System.Web.Services.Protocols.SoapDocumentMethodAttribute** or

20 **System.Web.Services.Protocols.SoapDocumentServiceAttribute** attribute to a

21 Web Service method or a method of a client proxy class and the class

22 implementing a Web Service respectively.

23 ToString

24

25 [C#] public const SoapParameterStyle Bare;

1 [C++] public: const SoapParameterStyle Bare;

2 [VB] Public Const Bare As SoapParameterStyle

3 [JScript] public var Bare : SoapParameterStyle;

4  
5 *Description*

6 Specifies the Web Service method parameters are elements directly beneath  
7 the **Body** element of the XML portion of a SOAP request or SOAP response.

8 ToString

9  
10 [C#] public const SoapParameterStyle Default;

11 [C++] public: const SoapParameterStyle Default;

12 [VB] Public Const Default As SoapParameterStyle

13 [JScript] public var Default : SoapParameterStyle;

14  
15 *Description*

16 Specifies using the default SoapParameterStyle for the Web Service. The  
17 default can for a Web Service by applying a SoapDocumentServiceAttribute to the  
18 class implementing the Web Service. If a

19 **System.Web.Services.Protocols.SoapDocumentServiceAttribute** is not applied  
20 to the class implementing the Web Service the default is

21 **System.Web.Services.Protocols.SoapParameterStyle.Wrapped** .

22 ToString

23  
24 [C#] public const SoapParameterStyle Wrapped;

25 [C++] public: const SoapParameterStyle Wrapped;

1 [VB] Public Const Wrapped As SoapParameterStyle

2 [JScript] public var Wrapped : SoapParameterStyle;

3  
4 *Description*

5 Specifies the Web Service method parameters are wrapped within a single  
6 element beneath the **Body** element of the XML portion of a SOAP request or  
7 SOAP response.

8 SoapRpcMethodAttribute class (System.Web.Services.Protocols)

9 ToString

10  
11  
12 *Description*

13 Applying the optional  
14 **System.Web.Services.Protocols.SoapRpcMethodAttribute** to a Web Service  
15 method alters the format of the SOAP request or response sent to and from a Web  
16 Service method.

17 Web Services Description Language (WSDL) defines two styles for how a  
18 Web Service method, which it calls an operation, can be encoded in a SOAP  
19 request or a SOAP response: RPC and Document. The RPC style refers to  
20 encoding the Web Service method according to the SOAP specification for using  
21 SOAP for RPC; otherwise known as Section 7 of the SOAP specification. This  
22 style states that all parameters are wrapped within a single element named after the  
23 Web Service method and that each element within that element represent a  
24 parameter named after their respective parameter name.

25 SoapRpcMethodAttribute

*Example Syntax:*

ToString

[C#] public SoapRpcMethodAttribute();

[C++] public: SoapRpcMethodAttribute();

[VB] Public Sub New()

[JScript] public function SoapRpcMethodAttribute(); Initializes a new instance of the **System.Web.Services.Protocols.SoapRpcMethodAttribute** class.

*Description*

Initializes a new instance of the **System.Web.Services.Protocols.SoapRpcMethodAttribute** class setting all properties to their defaults.

SoapRpcMethodAttribute

*Example Syntax:*

ToString

[C#] public SoapRpcMethodAttribute(string action);

[C++] public: SoapRpcMethodAttribute(String\* action);

[VB] Public Sub New(ByVal action As String)

[JScript] public function SoapRpcMethodAttribute(action : String);

*Description*

Initializes a new instance of the **System.Web.Services.Protocols.SoapRpcMethodAttribute** class setting the



1 **System.Web.Services.Protocols.SoapRpcMethodAttribute.Action** property to  
2 *action* . The intent of the SOAP request. Sets the

3 **System.Web.Services.Protocols.SoapRpcMethodAttribute.Action** property.

4 Action

5 ToString

6  
7 [C#] public string Action {get; set;}

8 [C++] public: \_\_property String\* get\_Action();public: \_\_property void

9 set\_Action(String\*);

10 [VB] Public Property Action As String

11 [JScript] public function get Action() : String;public function set Action(String);

12  
13 *Description*

14 Gets or sets the intent of the SOAP request.

15 The **System.Web.Services.Protocols.SoapRpcMethodAttribute.Action**  
16 property forms the SOAPAction HTTP Header Field for the HTTP Request.

17 Binding

18 ToString

19  
20 [C#] public string Binding {get; set;}

21 [C++] public: \_\_property String\* get\_Binding();public: \_\_property void

22 set\_Binding(String\*);

23 [VB] Public Property Binding As String

24 [JScript] public function get Binding() : String;public function set Binding(String);

1  
2 *Description*

3 Gets or sets the binding a Web Service method is implementing a operation  
4 for.

5 A binding, as defined by Web Services Description Language(WSDL), is  
6 similar to an interface, in that it defines a concrete set of operations. With respect  
7 to ASP.NET Web Services, each Web Service method is an operation within a  
8 binding. Web Service methods are members of either the default binding for a  
9 Web Service or in a binding specified within a

10 **System.Web.Services.WebServiceBindingAttribute** applied to a Web Service.

11 A Web Service can implement multiple bindings, by applying multiple

12 **System.Web.Services.WebServiceBindingAttribute** attributes to a Web Service.

13 OneWay

14 ToString

15  
16 [C#] public bool OneWay {get; set;}

17 [C++] public: \_\_property bool get\_OneWay();public: \_\_property void  
18 set\_OneWay(bool);

19 [VB] Public Property OneWay As Boolean

20 [JScript] public function get OneWay() : Boolean;public function set  
21 OneWay(Boolean);

22  
23 *Description*

24 Gets or sets whether a Web Service client waits for the Web server to finish  
25 processing a Web Service method.

When a Web Service method sets the **System.Web.Services.Protocols.SoapRpcMethodAttribute.OneWay** property to **true** , the Web Service client does not have to wait for the Web server to finish processing the Web Service method. As soon as the Web server has deserialized the **System.Web.Services.Protocols.SoapServerMessage** , but before invoking the Web Service method, the server returns an HTTP 202 status code. A HTTP 202 status code indicates to the client that the Web server has started processing of the message. Therefore, a Web Service client receives no acknowledgment that the Web server successfully processed the message.

RequestElementName

ToString

[C#] public string RequestElementName {get; set;}

[C++] public: \_\_property String\* get\_RequestElementName();public: \_\_property void set\_RequestElementName(String\*);

[VB] Public Property RequestElementName As String

[JScript] public function get RequestElementName() : String;public function set RequestElementName(String);

### *Description*

Gets or sets the XML element associated with the SOAP request for a Web Service method.

The

**System.Web.Services.Protocols.SoapRpcMethodAttribute.RequestElementNa**

1 **me** defines the XML element used to wrap the parameters beneath the **Body**  
2 element of the SOAP request.

3 RequestNamespace

4 ToString

5  
6 [C#] public string RequestNamespace {get; set;}

7 [C++] public: \_\_property String\* get\_RequestNamespace();public: \_\_property

8 void set\_RequestNamespace(String\*);

9 [VB] Public Property RequestNamespace As String

10 [JScript] public function get RequestNamespace() : String;public function set

11 RequestNamespace(String);

12  
13 *Description*

14 Gets or sets the XML namespace associated with the SOAP request for a  
15 Web Service method.

16 ResponseElementName

17 ToString

18  
19 [C#] public string ResponseElementName {get; set;}

20 [C++] public: \_\_property String\* get\_ResponseElementName();public: \_\_property

21 void set\_ResponseElementName(String\*);

22 [VB] Public Property ResponseElementName As String

23 [JScript] public function get ResponseElementName() : String;public function set

24 ResponseElementName(String);

1  
2 *Description*

3 Gets or sets the XML element associated with the SOAP response for a  
4 Web Service method.

5 The  
6 **System.Web.Services.Protocols.SoapRpcMethodAttribute.ResponseElement**  
7 **Name** defines the XML element used to wrap the parameters beneath the **Body**  
8 element of the SOAP response.

9 ResponseNamespace

10 ToString

11  
12 [C#] public string ResponseNamespace {get; set;}

13 [C++] public: \_\_property String\* get\_ResponseNamespace();public: \_\_property  
14 void set\_ResponseNamespace(String\*);

15 [VB] Public Property ResponseNamespace As String

16 [JScript] public function get ResponseNamespace() : String;public function set  
17 ResponseNamespace(String);

18  
19 *Description*

20 Gets or sets the XML namespace associated with the SOAP response for a  
21 Web Service method.

22 The  
23 **System.Web.Services.Protocols.SoapRpcMethodAttribute.ResponseNamespa**  
24 **ce** property is used in the XSD schema for the Web Service method in its Service  
25 Description.

1       TypeId

2       SoapRpcServiceAttribute class (System.Web.Services.Protocols)

3       ToString

4  
5  
6       *Description*

7           Applying the optional

8       **System.Web.Services.Protocols.SoapRpcServiceAttribute** to a Web Service  
9       sets the default format of SOAP requests and responses sent to and from Web  
10      Service methods within the Web Service.

11           The **System.Web.Services.Protocols.SoapRpcServiceAttribute** allows  
12      you to set the default encoding styles for Web Service methods within a Web  
13      Service to use RPC method encoding and  
14      **System.Web.Services.Description.SoapBindingUse.Encoded** parameter  
15      encoding.

16           SoapRpcServiceAttribute

17           *Example Syntax:*

18           ToString

19  
20      [C#] public SoapRpcServiceAttribute();

21      [C++] public: SoapRpcServiceAttribute();

22      [VB] Public Sub New()

23      [JScript] public function SoapRpcServiceAttribute();

24  
25       *Description*

1        Initializes a new instance of the  
2        **System.Web.Services.Protocols.SoapRpcServiceAttribute** class.

3        RoutingStyle

4        ToString

6        [C#] public SoapServiceRoutingStyle RoutingStyle {get; set;}

7        [C++] public: \_\_property SoapServiceRoutingStyle get\_RoutingStyle();public:

8        \_\_property void set\_RoutingStyle(SoapServiceRoutingStyle);

9        [VB] Public Property RoutingStyle As SoapServiceRoutingStyle

10        [JScript] public function get RoutingStyle() : SoapServiceRoutingStyle;public

11        function set RoutingStyle(SoapServiceRoutingStyle);

13        *Description*

15        TypeId

16        SoapServerMessage class (System.Web.Services.Protocols)

17        ToString

20        *Description*

21        Represents the data in a SOAP request received or a SOAP response sent  
22        by a Web Service method at a specific

23        **System.Web.Services.Protocols.SoapMessageStage** .

24        Action

25        ToString

```

1
2 [C#] public override string Action {get;}
3 [C++] public: __property virtual String* get_Action();
4 [VB] Overrides Public ReadOnly Property Action As String
5 [JScript] public function get Action() : String;
6

```

### *Description*

Gets the SOAPAction HTTP request header field for the SOAP request or SOAP response.

The **System.Web.Services.Protocols.SoapServerMessage.Action** property can be accessed during any

**System.Web.Services.Protocols.SoapMessageStage** .

ContentType

Exception

Headers

MethodInfo

ToString

### *Description*

Gets a representation of the method prototype for the Web Service method for which the SOAP request is intended.

Although the

**System.Web.Services.Protocols.SoapServerMessage.MethodInfo** property can be accessed during any stage, the method information is only available during



1 **System.Web.Services.Protocols.SoapMessageStage.AfterDeserialize** and  
2 **System.Web.Services.Protocols.SoapMessageStage.BeforeSerialize** .

3       OneWay

4       ToString

6 [C#] public override bool OneWay {get;}

7 [C++] public: \_\_property virtual bool get\_OneWay();

8 [VB] Overrides Public ReadOnly Property OneWay As Boolean

9 [JScript] public function get OneWay() : Boolean;

11 *Description*

12       Gets a value indicating whether the client waits for the server to finish  
13 processing a Web Service method.

14       A **System.Web.Services.Protocols.SoapDocumentMethodAttribute** or  
15 **System.Web.Services.Protocols.SoapRpcMethodAttribute** applied to a Web  
16 Service method or Web Service client can specify whether the Web Service  
17 method is one way or not by setting the

18 **System.Web.Services.Protocols.SoapDocumentMethodAttribute.OneWay**  
19 property of the attribute.

20       Server

21       ToString

23 [C#] public object Server {get;}

24 [C++] public: \_\_property Object\* get\_Server();

25 [VB] Public ReadOnly Property Server As Object

1 [JScript] public function get Server() : Object;

2  
3 *Description*

4 Gets the instance of the class handling the method invocation on the Web  
5 server **System.Web.Services.Protocols.SoapMessageStage** is not  
6 **System.Web.Services.Protocols.SoapMessageStage.AfterDeserialize** or  
7 **System.Web.Services.Protocols.SoapMessageStage.BeforeSerialize** .

8 The **System.Web.Services.Protocols.SoapServerMessage.Server**  
9 property is an instance of the class implementing the Web Service. If a SOAP  
10 extension knew ahead of time the type of the class, it could cast  
11 **System.Web.Services.Protocols.SoapServerMessage.Server** to that type and  
12 access properties and methods of the class implementing the Web Service.

13 Stage

14 Stream

15 Url

16 ToString

17  
18  
19 *Description*

20 Gets the base url of the Web Service.

21 The **System.Web.Services.Protocols.SoapServerMessage.Url** property  
22 can be accessed during any **System.Web.Services.Protocols.SoapMessageStage**

23  
24 EnsureInStage

1  
2 [C#] protected override void EnsureInStage();  
3 [C++] protected: void EnsureInStage();  
4 [VB] Overrides Protected Sub EnsureInStage()  
5 [JScript] protected override function EnsureInStage();  
6

7 *Description*

8 Asserts that the current  
9 **System.Web.Services.Protocols.SoapMessageStage** stage is a stage where in  
10 parameters are available. If not, an exception is thrown.

11 For Web Service methods, the in parameters are available after the SOAP  
12 request in the deserialization process in the  
13 **System.Web.Services.Protocols.SoapMessageStage.AfterDeserialize** stage. The  
14 **System.InvalidOperationException** is thrown if  
15 **System.Web.Services.Protocols.SoapServerMessage.EnsureInStage** method is  
16 invoked in any other **System.Web.Services.Protocols.SoapMessageStage** .

17 EnsureOutStage  
18

19 [C#] protected override void EnsureOutStage();  
20 [C++] protected: void EnsureOutStage();  
21 [VB] Overrides Protected Sub EnsureOutStage()  
22 [JScript] protected override function EnsureOutStage();  
23

24 *Description*  
25

1 Asserts that the current  
2 **System.Web.Services.Protocols.SoapMessageStage** stage is a stage where out  
3 parameters are available. If not, an exception is thrown.

4 For Web Service methods, the out parameters are available prior to the  
5 SOAP response serialization process in the  
6 **System.Web.Services.Protocols.SoapMessageStage.BeforeSerialize** stage. The  
7 **System.InvalidOperationException** is thrown if  
8 **System.Web.Services.Protocols.SoapServerMessage.EnsureOutStage** method  
9 is invoked in any other **System.Web.Services.Protocols.SoapMessageStage** .

10 SoapServiceRoutingStyle enumeration (System.Web.Services.Protocols)

11 ToString

12 ToString

13 ToString

14 SoapUnknownHeader class (System.Web.Services.Protocols)

15 ToString

16  
17  
18 *Description*

19 Represents the data received from a SOAP header that was not understood  
20 by the recipient Web Service or Web Service client. This class cannot be inherited.

21 A SOAP client may invoke a Web Service with additional data beyond just  
22 the required parameters in the form of a SOAP header. An ASP.NET Web Service  
23 or Web Service client can view any SOAP headers it did not know about at the  
24 time the Web Service was written by applying a

25 **System.Web.Services.Protocols.SoapHeaderAttribute** with a

**System.Web.Services.Protocols.SoapHeaderAttribute.MemberName** property of either an array of **System.Web.Services.Protocols.SoapHeader** , **System.Web.Services.Protocols.SoapHeader** , **System.Web.Services.Protocols.SoapUnknownHeader** or an array of **System.Web.Services.Protocols.SoapUnknownHeader** to the Web Service method. Specifying a **System.Type** of **System.Web.Services.Protocols.SoapUnknownHeader** , allows the Web Service to view the contents of the SOAP header and attempt to understand the semantics of the SOAP header, in the form of an **System.Xml.XmlElement** .

**SoapUnknownHeader**

*Example Syntax:*

**ToString**

[C#] public SoapUnknownHeader();

[C++] public: SoapUnknownHeader();

[VB] Public Sub New()

[JScript] public function SoapUnknownHeader();

**Actor**

**DidUnderstand**

**Element**

**ToString**

*Description*

Gets or sets the XML Header element for a SOAP request or response.

1 If a Web Service method wants to process SOAP headers it doesn't know  
2 about at the time the Web Service is written, a Web Service method can process an  
3 **System.Xml.XmlElement** class representing the raw XML of the SOAP header.

4 EncodedMustUnderstand

5 MustUnderstand

6 TextReturnReader class (System.Web.Services.Protocols)

7 ToString

8  
9  
10 *Description*

11  
12 TextReturnReader

13 *Example Syntax:*

14 ToString

15  
16 [C#] public TextReturnReader();

17 [C++] public: TextReturnReader();

18 [VB] Public Sub New()

19 [JScript] public function TextReturnReader();

20 GetInitializer

21  
22 [C#] public override object GetInitializer(LogicalMethodInfo methodInfo);

23 [C++] public: Object\* GetInitializer(LogicalMethodInfo\* methodInfo);

24 [VB] Overrides Public Function GetInitializer(ByVal methodInfo As

25 LogicalMethodInfo) As Object

1 [JScript] public override function GetInitializer(methodInfo : LogicalMethodInfo)  
2 : Object;

3  
4 *Description*

5  
6 Initialize

7  
8 [C#] public override void Initialize(object o);

9 [C++] public: void Initialize(Object\* o);

10 [VB] Overrides Public Sub Initialize(ByVal o As Object)

11 [JScript] public override function Initialize(o : Object);

12  
13 *Description*

14  
15 Read

16  
17 [C#] public override object Read(WebResponse response, Stream  
18 responseStream);

19 [C++] public: Object\* Read(WebResponse\* response, Stream\* responseStream);

20 [VB] Overrides Public Function Read(ByVal response As WebResponse, ByVal  
21 responseStream As Stream) As Object

22 [JScript] public override function Read(response : WebResponse, responseStream  
23 : Stream) : Object;

24  
25 *Description*

1  
2       UrlEncodedParameterWriter class (System.Web.Services.Protocols)

3       ToString  
4  
5

6       *Description*  
7

8       UrlEncodedParameterWriter

9       *Example Syntax:*

10      ToString  
11

12     [C#] protected UrlEncodedParameterWriter();

13     [C++] protected: UrlEncodedParameterWriter();

14     [VB] Protected Sub New()

15     [JScript] protected function UrlEncodedParameterWriter();

16       RequestEncoding

17       ToString  
18

19     [C#] public override Encoding RequestEncoding {get; set;}

20     [C++] public: \_\_property virtual Encoding\* get\_RequestEncoding();public:

21     \_\_property virtual void set\_RequestEncoding(Encoding\*);

22     [VB] Overrides Public Property RequestEncoding As Encoding

23     [JScript] public function get RequestEncoding() : Encoding;public function set

24     RequestEncoding(Encoding);  
25



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

*Description*

UsesWriteRequest  
Encode

[C#] protected void Encode(TextWriter writer, object[] values);  
[C++] protected: void Encode(TextWriter\* writer, Object\* values \_\_gc[]);  
[VB] Protected Sub Encode(ByVal writer As TextWriter, ByVal values() As  
Object)  
[JScript] protected function Encode(writer : TextWriter, values : Object[]);

*Description*

Encode

[C#] protected void Encode(TextWriter writer, string name, object value);  
[C++] protected: void Encode(TextWriter\* writer, String\* name, Object\* value);  
[VB] Protected Sub Encode(ByVal writer As TextWriter, ByVal name As String,  
ByVal value As Object)  
[JScript] protected function Encode(writer : TextWriter, name : String, value :  
Object);

*Description*

1           GetInitializer

2

3   [C#] public override object GetInitializer(LogicalMethodInfo methodInfo);

4   [C++] public: Object\* GetInitializer(LogicalMethodInfo\* methodInfo);

5   [VB] Overrides Public Function GetInitializer(ByVal methodInfo As

6   LogicalMethodInfo) As Object

7   [JScript] public override function GetInitializer(methodInfo : LogicalMethodInfo)

8   : Object;

9

10   *Description*

11

12           Initialize

13

14   [C#] public override void Initialize(object initializer);

15   [C++] public: void Initialize(Object\* initializer);

16   [VB] Overrides Public Sub Initialize(ByVal initializer As Object)

17   [JScript] public override function Initialize(initializer : Object);

18

19   *Description*

20

21           UrlParameterReader class (System.Web.Services.Protocols)

22           WriteRequest

23

24

25   *Description*

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

UrlParameterReader

*Example Syntax:*

WriteRequest

```
[C#] public UrlParameterReader();  
[C++] public: UrlParameterReader();  
[VB] Public Sub New()  
[JScript] public function UrlParameterReader();  
  
Read  
  
[C#] public override object[] Read(HttpRequest request);  
[C++] public: Object* Read(HttpRequest* request) __gc[];  
[VB] Overrides Public Function Read(ByVal request As HttpRequest) As Object()  
[JScript] public override function Read(request : HttpRequest) : Object[];
```

*Description*

UrlParameterWriter class (System.Web.Services.Protocols)

ToString

*Description*

UrlParameterWriter

*Example Syntax:*

ToString

[C#] public UrlParameterWriter();

[C++] public: UrlParameterWriter();

[VB] Public Sub New()

[JScript] public function UrlParameterWriter();

RequestEncoding

UsesWriteRequest

GetRequestUrl

[C#] public override string GetRequestUrl(string url, object[] parameters);

[C++] public: String\* GetRequestUrl(String\* url, Object\* parameters \_\_gc[]);

[VB] Overrides Public Function GetRequestUrl(ByVal url As String, ByVal

parameters() As Object) As String

[JScript] public override function GetRequestUrl(url : String, parameters :

Object[]) : String;

*Description*

ValueCollectionParameterReader class (System.Web.Services.Protocols)

WriteRequest

*Description*

ValueCollectionParameterReader

*Example Syntax:*

WriteRequest

[C#] protected ValueCollectionParameterReader();

[C++] protected: ValueCollectionParameterReader();

[VB] Protected Sub New()

[JScript] protected function ValueCollectionParameterReader();

GetInitializer

[C#] public override object GetInitializer(LogicalMethodInfo methodInfo);

[C++] public: Object\* GetInitializer(LogicalMethodInfo\* methodInfo);

[VB] Overrides Public Function GetInitializer(ByVal methodInfo As

LogicalMethodInfo) As Object

[JScript] public override function GetInitializer(methodInfo : LogicalMethodInfo)

: Object;

*Description*

Initialize

[C#] public override void Initialize(object o);

[C++] public: void Initialize(Object\* o);

[VB] Overrides Public Sub Initialize(ByVal o As Object)

1 [JScript] public override function Initialize(o : Object);

3 *Description*

5       IsSupported

7 [C#] public static bool IsSupported(LogicalMethodInfo methodInfo);

8 [C++] public: static bool IsSupported(LogicalMethodInfo\* methodInfo);

9 [VB] Public Shared Function IsSupported(ByVal methodInfo As

10 LogicalMethodInfo) As Boolean

11 [JScript] public static function IsSupported(methodInfo : LogicalMethodInfo) :

12 Boolean;

14 *Description*

16       IsSupported

18 [C#] public static bool IsSupported(ParameterInfo paramInfo);

19 [C++] public: static bool IsSupported(ParameterInfo\* paramInfo);

20 [VB] Public Shared Function IsSupported(ByVal paramInfo As ParameterInfo) As

21 Boolean

22 [JScript] public static function IsSupported(paramInfo : ParameterInfo) : Boolean;

24 *Description*

1       Read

2

3   [C#] protected object[] Read(NameValueCollection collection);

4   [C++] protected: Object\* Read(NameValueCollection\* collection) \_\_gc[];

5   [VB] Protected Function Read(ByVal collection As NameValueCollection) As

6   Object()

7   [JScript] protected function Read(collection : NameValueCollection) : Object[];

8

9       *Description*

10

11       WebClientAsyncResult class (System.Web.Services.Protocols)

12       ToString

13

14

15       *Description*

16

17       AsyncState

18       ToString

19

20   [C#] public object AsyncState {get;}

21   [C++] public: \_\_property Object\* get\_AsyncState();

22   [VB] Public ReadOnly Property AsyncState As Object

23   [JScript] public function get AsyncState() : Object;

24

25       *Description*

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

AsyncWaitHandle

ToString

[C#] public WaitHandle AsyncWaitHandle {get;}

[C++] public: \_\_property WaitHandle\* get\_AsyncWaitHandle();

[VB] Public ReadOnly Property AsyncWaitHandle As WaitHandle

[JScript] public function get AsyncWaitHandle() : WaitHandle;

*Description*

CompletedSynchronously

ToString

[C#] public bool CompletedSynchronously {get;}

[C++] public: \_\_property bool get\_CompletedSynchronously();

[VB] Public ReadOnly Property CompletedSynchronously As Boolean

[JScript] public function get CompletedSynchronously() : Boolean;

*Description*

Gets a value indicating whether the invocation of the Web Service method completed synchronously.

IsCompleted

ToString



1  
2 [C#] public bool IsCompleted {get;}

3 [C++] public: \_\_property bool get\_IsCompleted();

4 [VB] Public ReadOnly Property IsCompleted As Boolean

5 [JScript] public function get IsCompleted() : Boolean;

6  
7 *Description*

8 Gets a value indicating whether the asynchronous request has completed.

9 **Abort**

10  
11 [C#] public void Abort();

12 [C++] public: void Abort();

13 [VB] Public Sub Abort()

14 [JScript] public function Abort();

15  
16 *Description*

17  
18 WebClientProtocol class (System.Web.Services.Protocols)

19 **ToString**

20  
21  
22 *Description*

23 Specifies the base class for all ASP.NET Web Service client proxies.

24 ASP.NET incorporates two distinct functionalities of Web Services:

25 building ASP.NET Web Services and building Web Service clients. If you are

building a Web Service client using ASP.NET, you must create a proxy class deriving indirectly or directly from **System.Web.Services.Protocols.WebClientProtocol** for the Web Service you want to call.

**WebClientProtocol**

*Example Syntax:*

**ToString**

[C#] protected WebClientProtocol();

[C++] protected: WebClientProtocol();

[VB] Protected Sub New()

[JScript] protected function WebClientProtocol();

#### *Description*

Initializes a new instance of the **System.Web.Services.Protocols.WebClientProtocol** class.

**ConnectionGroupName**

**ToString**

[C#] public string ConnectionGroupName {get; set;}

[C++] public: \_\_property String\* get\_ConnectionGroupName();public: \_\_property void set\_ConnectionGroupName(String\*);

[VB] Public Property ConnectionGroupName As String

[JScript] public function get ConnectionGroupName() : String;public function set ConnectionGroupName(String);

1  
2 *Description*

3 Gets or sets the name of the connection group for the request.

4 The

5 **System.Web.Services.Protocols.WebClientProtocol.ConnectionGroupName**

6 property enables you to associate a request to a connection group. For more

7 details, see **System.Net.HttpWebRequest.ConnectionGroupName** .

8 Container

9 Credentials

10 ToString

11  
12  
13 *Description*

14 Gets or sets security credentials for Web Service client authentication.

15 When using the **Credentials** property, a Web Service client must  
16 instantiate a class implementing **System.Net.ICredentials** , such as  
17 **System.Net.NetworkCredential** , and then set the client credentials specific to  
18 the authentication mechanism. The **System.Net.NetworkCredential** class can be  
19 used to set authentication credentials using the basic, digest, NTLM and Kerberos  
20 authentication mechanisms.

21 DesignMode

22 Events

23 PreAuthenticate

24 ToString

1  
2  
3 *Description*

4 Gets or sets whether pre-authentication is enabled.

5 When

6 **System.Web.Services.Protocols.WebClientProtocol.PreAuthenticate** is **true** ,  
7 the **WWW-authenticate** header is sent with each request back to the server,  
8 otherwise standard authentication procedures are used. When  
9 **System.Web.Services.Protocols.WebClientProtocol.PreAuthenticate** is **false** ,  
10 the authentication headers are sent only on the first request.

11 RequestEncoding

12 ToString

13  
14 [C#] public Encoding RequestEncoding {get; set;}

15 [C++] public: \_\_property Encoding\* get\_RequestEncoding();public: \_\_property  
16 void set\_RequestEncoding(Encoding\*);

17 [VB] Public Property RequestEncoding As Encoding

18 [JScript] public function get RequestEncoding() : Encoding;public function set  
19 RequestEncoding(Encoding);

20  
21 *Description*

22 The **System.Text.Encoding** used to make the client request to the Web  
23 Service.

24 The

25 **System.Web.Services.Protocols.WebClientProtocol.RequestEncoding**

determines the encoding for the request message. The

**System.Net.WebRequest.ContentType** of the request will be annotated with the encoding value.

Site

Timeout

ToString

### *Description*

Indicates the time a Web Service client waits for a synchronous Web Service request to complete (in milliseconds).

Even though a Web Service client can set the **System.Web.Services.Protocols.WebClientProtocol.Timeout** property to infinity with a value of -1, the Web Server can still cause the request to time out on the server side.

Url

ToString

```
[C#] public string Url {get; set;}
```

```
[C++] public: __property String* get_Url();public: __property void  
set_Url(String*);
```

```
[VB] Public Property Url As String
```

```
[JScript] public function get Url() : String;public function set Url(String);
```

### *Description*

1 Gets or sets the base URL of the Web Service the client is requesting.

2 Proxy classes generated using Wsdll.exe will set a default

3 **System.Web.Services.Protocols.WebClientProtocol.Url** property for the client  
4 to use. The default **System.Web.Services.Protocols.WebClientProtocol.Url** is  
5 determined by the location attribute found in the Service Description from which  
6 the proxy class was generated.

## 7 Abort

8  
9 [C#] public virtual void Abort();

10 [C++] public: virtual void Abort();

11 [VB] Overridable Public Sub Abort()

12 [JScript] public function Abort();

## 13 14 *Description*

15 Cancels a request to a Internet resource.

16 *Return Value:* **System.Web.Services.Protocols.WebClientProtocol.Abort**

17 cancels a request to a resource made using the

18 **System.Web.Services.Protocols.WebClientProtocol.GetWebRequest(System.**

19 **Uri)** method. After the request is canceled, invoking

20 **System.Web.Services.Protocols.WebClientProtocol.GetResponse(System.**

21 **Net.WebRequest)** will cause a **System.Net.WebException** .

## 22 AddToCache

23  
24 [C#] protected static void AddToCache(Type type, object value);

25 [C++] protected: static void AddToCache(Type\* type, Object\* value);

1 [VB] Protected Shared Sub AddToCache(ByVal type As Type, ByVal value As  
2 Object)

3 [JScript] protected static function AddToCache(type : Type, value : Object);

4  
5 *Description*

6 Add an instance of the client protocol handler to the cache.

7 GetFromCache

8  
9 [C#] protected static object GetFromCache(Type type);

10 [C++] protected: static Object\* GetFromCache(Type\* type);

11 [VB] Protected Shared Function GetFromCache(ByVal type As Type) As Object

12 [JScript] protected static function GetFromCache(type : Type) : Object;

13  
14 *Description*

15 Gets an instance of a client protocol handler from the cache.

16 GetWebRequest

17  
18 [C#] protected virtual WebRequest GetWebRequest(Uri uri);

19 [C++] protected: virtual WebRequest\* GetWebRequest(Uri\* uri);

20 [VB] Overridable Protected Function GetWebRequest(ByVal uri As Uri) As  
21 WebRequest

22 [JScript] protected function GetWebRequest(uri : Uri) : WebRequest;

23  
24 *Description*

Creates a **System.Net.WebRequest** instance for the specified *url* .

*Return Value:* The **System.Net.WebRequest** instance.

This method can be overridden to customize the **System.Net.WebRequest** object before the Web Service request is made. For example you could add a custom header to the request. The **System.Uri** to use when creating the **System.Net.WebRequest**.

GetWebResponse

[C#] protected virtual WebResponse GetWebResponse(WebRequest request);

[C++] protected: virtual WebResponse\* GetWebResponse(WebRequest\* request);

[VB] Overridable Protected Function GetWebResponse(ByVal request As WebRequest) As WebResponse

[JScript] protected function GetWebResponse(request : WebRequest) :

WebResponse; Returns a response from a request to a Web Service method.

#### *Description*

Returns a response from a synchronous request to a Web Service method.

*Return Value:* A **System.Net.WebResponse** . The **System.Net.WebRequest** to get the response from.

GetWebResponse

[C#] protected virtual WebResponse GetWebResponse(WebRequest request, IAsyncResult result);

[C++] protected: virtual WebResponse\* GetWebResponse(WebRequest\* request, IAsyncResult\* result);



```

1 [VB] Overridable Protected Function GetWebResponse(ByVal request As
2 WebRequest, ByVal result As IAsyncResult) As WebResponse
3 [JScript] protected function GetWebResponse(request : WebRequest, result :
4 IAsyncResult) : WebResponse;

```

#### *Description*

Returns a response from an asynchronous request to a Web Service method.

*Return Value:* The **System.Net.WebResponse** instance. The **System.Net.WebRequest** to get the response from. The **System.IAsyncResult** to pass to **System.Net.HttpWebRequest.EndGetResponse(System.IAsyncResult)** when the response has completed.

WebServiceHandlerFactory class (System.Web.Services.Protocols)

ToString

#### *Description*

WebServiceHandlerFactory

*Example Syntax:*

ToString

```

22 [C#] public WebServiceHandlerFactory();
23 [C++] public: WebServiceHandlerFactory();
24 [VB] Public Sub New()
25 [JScript] public function WebServiceHandlerFactory();

```

## GetHandler

[C#] public IHttpHandler GetHandler(HttpContext context, string verb, string url, string filePath);

[C++] public: \_\_sealed IHttpHandler\* GetHandler(HttpContext\* context, String\* verb, String\* url, String\* filePath);

[VB] NotOverridable Public Function GetHandler(ByVal context As HttpContext, ByVal verb As String, ByVal url As String, ByVal filePath As String) As IHttpHandler

[JScript] public function GetHandler(context : HttpContext, verb : String, url : String, filePath : String) : IHttpHandler;

### *Description*

## ReleaseHandler

[C#] public void ReleaseHandler(IHttpHandler handler);

[C++] public: \_\_sealed void ReleaseHandler(IHttpHandler\* handler);

[VB] NotOverridable Public Sub ReleaseHandler(ByVal handler As IHttpHandler)

[JScript] public function ReleaseHandler(handler : IHttpHandler);

### *Description*

XmlReturnReader class (System.Web.Services.Protocols)

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

ToString

*Description*

XmlReturnReader

*Example Syntax:*

ToString

[C#] public XmlReturnReader();

[C++] public: XmlReturnReader();

[VB] Public Sub New()

[JScript] public function XmlReturnReader();

GetInitializer

[C#] public override object GetInitializer(LogicalMethodInfo methodInfo);

[C++] public: Object\* GetInitializer(LogicalMethodInfo\* methodInfo);

[VB] Overrides Public Function GetInitializer(ByVal methodInfo As

LogicalMethodInfo) As Object

[JScript] public override function GetInitializer(methodInfo : LogicalMethodInfo)

: Object;

*Description*

GetInitializers

```

1
2 [C#] public override object[] GetInitializers(LogicalMethodInfo[] methodInfos);
3 [C++] public: Object* GetInitializers(LogicalMethodInfo* methodInfos[]) __gc[];
4 [VB] Overrides Public Function GetInitializers(ByVal methodInfos() As
5 LogicalMethodInfo) As Object()
6 [JScript] public override function GetInitializers(methodInfos :
7 LogicalMethodInfo[]) : Object[];
8

```

### *Description*

#### Initialize

```

13 [C#] public override void Initialize(object o);
14 [C++] public: void Initialize(Object* o);
15 [VB] Overrides Public Sub Initialize(ByVal o As Object)
16 [JScript] public override function Initialize(o : Object);
17

```

### *Description*

#### Read

```

22 [C#] public override object Read(WebResponse response, Stream
23 responseStream);
24
25

```

## System.Web.SessionState

### *Description*

HttpSessionState class (System.Web.SessionState)

### *Description*

Provides access to session state values as well as session-level settings and lifetime management methods.

Properties:

CodePage

[C#] public int CodePage {get; set;}

[C++] public: \_\_property int get\_CodePage();public: \_\_property void  
set\_CodePage(int);

[VB] Public Property CodePage As Integer

[JScript] public function get CodePage() : int;public function set CodePage(int);

### *Description*

Gets or sets the code page identifier for the current session.

Contents

[C#] public HttpSessionState Contents {get;}

1 [C++] public: \_\_property HttpSessionState\* get\_Contents();  
2 [VB] Public ReadOnly Property Contents As HttpSessionState  
3 [JScript] public function get Contents() : HttpSessionState;

4  
5 *Description*

6 Gets a reference to the current session state object.

7 **Contents** is provided for legacy ASP compatibility.

8 **Count**

9  
10 [C#] public int Count {get;}  
11 [C++] public: \_\_property int get\_Count();  
12 [VB] Public ReadOnly Property Count As Integer  
13 [JScript] public function get Count() : int;

14  
15 *Description*

16 Gets the number of items in the session state collection.

17 **IsCookieless**

18  
19 [C#] public bool IsCookieless {get;}  
20 [C++] public: \_\_property bool get\_IsCookieless();  
21 [VB] Public ReadOnly Property IsCookieless As Boolean  
22 [JScript] public function get IsCookieless() : Boolean;

23  
24 *Description*  
25

1 Gets a value indicating whether the session is managed using cookieless  
2 session.

### 3 IsNewSession

4  
5 [C#] public bool IsNewSession {get;}

6 [C++] public: \_\_property bool get\_IsNewSession();

7 [VB] Public ReadOnly Property IsNewSession As Boolean

8 [JScript] public function get IsNewSession() : Boolean;

### 9 10 *Description*

11 Gets a value indicating whether the session has been created with the  
12 current request.

### 13 IsReadOnly

14  
15 [C#] public bool IsReadOnly {get;}

16 [C++] public: \_\_property bool get\_IsReadOnly();

17 [VB] Public ReadOnly Property IsReadOnly As Boolean

18 [JScript] public function get IsReadOnly() : Boolean;

### 19 20 *Description*

21 Gets a value indicating whether the session is read-only.

### 22 IsSynchronized

23  
24 [C#] public bool IsSynchronized {get;}

25 [C++] public: \_\_property bool get\_IsSynchronized();

1 [VB] Public ReadOnly Property IsSynchronized As Boolean

2 [JScript] public function get IsSynchronized() : Boolean;

3  
4 *Description*

5 Gets a value indicating whether access to the collection of session state  
6 values is read-only (thread-safe).

7 *Item*

8  
9 [C#] public object this[string name] {get; set;}

10 [C++] public: \_\_property Object\* get\_Item(String\* name);public: \_\_property void  
11 set\_Item(String\* name, Object\*);

12 [VB] Public Default Property Item(ByVal name As String) As Object

13 [JScript] returnValue =

14 HttpSessionStateObject.Item(name);HttpSessionStateObject.Item(name) =

15 returnValue; Gets or sets individual session values.

16  
17 *Description*

18 Gets or sets a session value by name. The key name of the session value.

19 *Item*

20  
21 [C#] public object this[int index] {get; set;}

22 [C++] public: \_\_property Object\* get\_Item(int index);public: \_\_property void  
23 set\_Item(int index, Object\*);

24 [VB] Public Default Property Item(ByVal index As Integer) As Object

25 [JScript] returnValue =



1 HttpSessionStateObject.Item(index);HttpSessionStateObject.Item(index) =  
2 return Value;

3  
4 *Description*

5 Gets or sets a session value by numerical index. The numerical index of the  
6 session value.

7 Keys

8  
9 [C#] public NameObjectCollectionBase.KeysCollection Keys {get;}

10 [C++] public: \_\_property NameObjectCollectionBase.KeysCollection\*

11 get\_Keys();

12 [VB] Public ReadOnly Property Keys As

13 NameObjectCollectionBase.KeysCollection

14 [JScript] public function get Keys() : NameObjectCollectionBase.KeysCollection;

15  
16 *Description*

17 Gets a collection of all session keys.

18 LCID

19  
20 [C#] public int LCID {get; set;}

21 [C++] public: \_\_property int get\_LCID();public: \_\_property void set\_LCID(int);

22 [VB] Public Property LCID As Integer

23 [JScript] public function get LCID() : int;public function set LCID(int);

24  
25 *Description*

Gets or sets the locale identifier (LCID) of the current session.

Mode

[C#] public SessionStateMode Mode {get;}

[C++] public: \_\_property SessionStateMode get\_Mode();

[VB] Public ReadOnly Property Mode As SessionStateMode

[JScript] public function get Mode() : SessionStateMode;

*Description*

Gets the current session state mode.

SessionID

[C#] public string SessionID {get;}

[C++] public: \_\_property String\* get\_SessionID();

[VB] Public ReadOnly Property SessionID As String

[JScript] public function get SessionID() : String;

*Description*

Gets the unique session ID used to identify a session.

StaticObjects

[C#] public HttpStaticObjectsCollection StaticObjects {get;}

[C++] public: \_\_property HttpStaticObjectsCollection\* get\_StaticObjects();

[VB] Public ReadOnly Property StaticObjects As HttpStaticObjectsCollection

[JScript] public function get StaticObjects() : HttpStaticObjectsCollection;

1  
2 *Description*

3 Gets a collection of objects declared by  
4  
5  
6

7 **System.Web.UI**  
8

9 *Description*

10 The System.Web.UI namespace provides classes and interfaces that allow  
11 you to create controls and pages that will appear in your Web applications as user  
12 interface on a Web page. This namespace includes the Control class, which  
13 provides all controls, whether HTML, Web, or User controls, with a common set  
14 of functionality. It also includes the Page control, which is generated automatically  
15 whenever a request is made for a page in your Web application. Also provided are  
16 classes which provide the Web Forms Server Controls data binding functionality,  
17 the ability to save the view state of a given control or page, as well as parsing  
18 functionality for both programmable and literal controls.

19 AttributeCollection class (System.Web.UI)  
20  
21

22 *Description*

23 Provides object-model access to all attributes declared in the opening tag of  
24 an ASP.NET server control element. This class cannot be inherited.  
25

Attributes are case-insensitive strings. They return a **System.String** object as their value. If there are no attributes in the collection, they return **null**.

Constructors:

AttributeCollection

*Example Syntax:*

```
[C#]          public          AttributeCollection(StateBag          bag);
[C++]          public:          AttributeCollection(StateBag*          bag);
[VB]      Public      Sub      New(ByVal      bag      As      StateBag)
[JScript]      public      function      AttributeCollection(bag      :      StateBag);
```

*Description*

Initializes a new instance of the **System.Web.UI.AttributeCollection** class. A **System.Web.UI.StateBag** object that contains the attribute keys and their values that are in the opening tag of the server control.

Properties:

Count

```
[C#]          public          int          Count          {get;}
[C++]          public:          __property          int          get_Count();
[VB]      Public      ReadOnly      Property      Count      As      Integer
[JScript]      public      function      get      Count()      :      int;
```

*Description*

1 Gets the number of attributes in the **System.Web.UI.AttributeCollection**  
2 object.

3 **CssStyle**

4  
5 [C#] public **CssStyleCollection** **CssStyle** {get;}  
6 [C++] public: \_\_property **CssStyleCollection\*** get\_CssStyle();  
7 [VB] Public ReadOnly Property **CssStyle** As **CssStyleCollection**  
8 [JScript] public function get **CssStyle()** : **CssStyleCollection**;

9  
10 *Description*

11 Gets a collection of styles for the ASP.NET server control to which the  
12 current **System.Web.UI.AttributeCollection** object belongs.

13 Any style declared for a particular HTML server control is added to the  
14 **CssStyleCollection** object when the containing Web Forms page is parsed. You  
15 can use this property to add, remove and iterate through the styles declared for the  
16 server control.

17 **Item**

18  
19 [C#] public string **this**[string **key**] {get; set;}  
20 [C++] public: \_\_property String\* get\_Item(String\* **key**);public: \_\_property void  
21 set\_Item(String\* **key**, String\*);  
22 [VB] Public Default Property **Item**(ByVal **key** As String) As String  
23 [JScript] **returnValue** =  
24 **AttributeCollectionObject.Item(key);AttributeCollectionObject.Item(key)** =  
25 **returnValue**;

1  
2 *Description*

3 Gets or sets a specified attribute value for a server control. The location of  
4 the attribute within the collection.

5 Keys

6  
7 [C#] public ICollection Keys {get;}  
8 [C++] public: \_\_property ICollection\* get\_Keys();  
9 [VB] Public ReadOnly Property Keys As ICollection  
10 [JScript] public function get Keys() : ICollection;  
11

12 *Description*

13 Gets a collection of keys to all attributes in the server control's  
14 **System.Web.UI.AttributeCollection** object.

15 Methods:

16 Add

17  
18 [C#] public void Add(string key, string value);  
19 [C++] public: void Add(String\* key, String\* value);  
20 [VB] Public Sub Add(ByVal key As String, ByVal value As String)  
21 [JScript] public function Add(key : String, value : String);  
22

23 *Description*  
24  
25

1 Adds an attribute to a server control's **System.Web.UI.AttributeCollection**  
2 object. The index assigned to the new attribute in the collection. The attribute to  
3 store in the collection.

#### 4 AddAttributes

5  
6 [C#] public void AddAttributes(HtmlTextWriter writer);  
7 [C++] public: void AddAttributes(HtmlTextWriter\* writer);  
8 [VB] Public Sub AddAttributes(ByVal writer As HtmlTextWriter)  
9 [JScript] public function AddAttributes(writer : HtmlTextWriter);  
10

#### 11 *Description*

12 Adds attributes from the **AttributeCollection** class to the  
13 **System.Web.UI.HtmlTextWriter** object that is responsible for rendering the  
14 attributes as HTML to an ASP.NET server control.

15 This method copies all of the server control's attributes to an  
16 **System.Web.UI.HtmlTextWriter** object so they can be rendered by the next call  
17 to the **System.Web.UI.HtmlTextWriter.RenderBeginTag(System.String)**  
18 method. An **System.Web.UI.HtmlTextWriter** that writes the added attribute to  
19 the opening tag of an ASP.NET server control.

#### 20 Clear

21  
22 [C#] public void Clear();  
23 [C++] public: void Clear();  
24 [VB] Public Sub Clear()  
25 [JScript] public function Clear();

*Description*

Removes all attributes from a server control's **System.Web.UI.AttributeCollection** object.

Remove

[C#]            public            void            Remove(string            key);

[C++]            public:            void            Remove(String\*            key);

[VB]    Public    Sub    Remove(ByVal    key    As    String)

[JScript]    public    function    Remove(key    :    String);

*Description*

Removes an attribute from a server control's **System.Web.UI.AttributeCollection** object. The index of the attribute to remove.

Render

[C#]            public            void            Render(HtmlTextWriter            writer);

[C++]            public:            void            Render(HtmlTextWriter\*            writer);

[VB]    Public    Sub    Render(ByVal    writer    As    HtmlTextWriter)

[JScript]    public    function    Render(writer    :    HtmlTextWriter);

*Description*

Writes the collection of attributes to the specified **System.Web.UI.HtmlTextWriter** output stream. In turn, the output stream writes



the collection to the Web Forms page. An **System.Web.UI.HtmlTextWriter** object that writes the attribute collection to the opening HTML tag.

AutomaticHandlerMethodInfos class (System.Web.UI)

ToString

#### *Description*

AutomaticHandlerMethodInfos

#### *Example Syntax:*

ToString

[C#]                      public                      AutomaticHandlerMethodInfos();

[C++]                    public:                    AutomaticHandlerMethodInfos();

[VB]                    Public                    Sub                    New()

[JScript] public function AutomaticHandlerMethodInfos();

BaseParser class (System.Web.UI)

ToString

#### *Description*

Provides a base set of functionality for all parsers involved in parsing ASP.NET page requests. This includes parsing of all ASP.NET server controls, including pages and user controls.

BaseParser

#### *Example Syntax:*

```

1      ToString
2
3      [C#]                public                BaseParser();
4      [C++]              public:                BaseParser();
5      [VB]                Public                Sub                New()
6      [JScript] public function BaseParser();

```

```

7      BasePartialCachingControl class (System.Web.UI)
8      ToString
9
10

```

11 *Description*

```

12      Provides a base set of functionality for the
13      System.Web.UI.StaticPartialCachingControl and the
14      System.Web.UI.PartialCachingControl classes.

```

```

15      BasePartialCachingControl

```

16 *Example Syntax:*

```

17      ToString
18

```

```

19      [C#]                protected                BasePartialCachingControl();
20      [C++]              protected:                BasePartialCachingControl();
21      [VB]                Protected                Sub                New()
22      [JScript] protected function BasePartialCachingControl();

```

```

23      ChildControlsCreated

```

```

24      ClientID

```

```

25      Context

```

1	Controls				
2	Dependency				
3	ToString				
4					
5					
6	<i>Description</i>				
7	Gets or sets an instance of the <b>System.Web.Caching.CacheDependency</b>				
8	class associated with the cached server control output.				
9	EnableViewState				
10	Events				
11	HasChildViewState				
12	ID				
13	IsTrackingViewState				
14	NamingContainer				
15	Page				
16	Parent				
17	Site				
18	TemplateSourceDirectory				
19	UniqueID				
20	ViewState				
21	ViewStateIgnoresCase				
22	Visible				
23	Dispose				
24					
25	[C#]	public	override	void	Dispose();

```

1 [C++]          public:          void          Dispose();
2 [VB]           Overrides          Public          Sub          Dispose()
3 [JScript]      public          override          function          Dispose();

```

#### *Description*

### OnInit

```

9 [C#]           protected          override          void          OnInit(EventArgs          e);
10 [C++]          protected:          void          OnInit(EventArgs*          e);
11 [VB]           Overrides          Protected          Sub          OnInit(ByVal          e          As          EventArgs)
12 [JScript]      protected          override          function          OnInit(e          :          EventArgs);

```

#### *Description*

Raises the **Init** event for the server control to be output cached.

This method checks the cache for the content associated with the server control. If it is not there, the server control is created and added to the parent control's **System.Web.UI.ControlCollection** object. An **System.EventArgs** object that contains the event data.

### Render

```

22 [C#]           protected          override          void          Render(HtmlTextWriter          output);
23 [C++]          protected:          void          Render(HtmlTextWriter*          output);
24 [VB]           Overrides          Protected          Sub          Render(ByVal          output          As          HtmlTextWriter)
25 [JScript]      protected          override          function          Render(output          :          HtmlTextWriter);

```

1  
2 *Description*

3       Outputs       the       server       control's       content       to       the  
4 **System.Web.UI.HtmlTextWriter**       output       stream.       The  
5 **System.Web.UI.HtmlTextWriter** object that writes the cached control to the  
6 page.

7       BuildMethod delegate (System.Web.UI)

8       TrackViewState

9  
10  
11 *Description*

12       Represents the method that is used to build a **System.Web.UI.Control** .

13       This delegate passes the information necessary to build ASP.NET server  
14 controls as defined by the TemplateControl class, or one of the classes that inherit  
15 from TemplateControl, Page and UserControl.

16       BuildTemplateMethod delegate (System.Web.UI)

17       TrackViewState

18  
19  
20 *Description*

21  
22       CompiledTemplateBuilder class (System.Web.UI)

23       TrackViewState

1  
2  
3 *Description*  
4

5 CompiledTemplateBuilder

6 *Example Syntax:*

7 TrackViewState  
8

9 [C#] public CompiledTemplateBuilder(BuildTemplateMethod  
10 buildTemplateMethod);

11 [C++] public: CompiledTemplateBuilder(BuildTemplateMethod\*  
12 buildTemplateMethod);

13 [VB] Public Sub New(ByVal buildTemplateMethod As BuildTemplateMethod)

14 [JScript] public function CompiledTemplateBuilder(buildTemplateMethod :  
15 BuildTemplateMethod);  
16

17 *Description*  
18

19 InstantiateIn  
20

21 [C#] public void InstantiateIn(Control container);

22 [C++] public: \_\_sealed void InstantiateIn(Control\* container);

23 [VB] NotOverridable Public Sub InstantiateIn(ByVal container As Control)

24 [JScript] public function InstantiateIn(container : Control);  
25

1  
2 *Description*

3  
4 ConstructorNeedsTagAttribute class (System.Web.UI)  
5 ToString  
6  
7

8 *Description*

9 Specifies that a server control needs a tag name in its constructor.

10 ConstructorNeedsTagAttribute

11 *Example Syntax:*

12 ToString  
13

14 [C#] public ConstructorNeedsTagAttribute();  
15 [C++] public: ConstructorNeedsTagAttribute();  
16 [VB] Public Sub New()  
17 [JScript] public function ConstructorNeedsTagAttribute(); Initializes a new  
18 instance of the **System.Web.UI.ConstructorNeedsTagAttribute** class.  
19

20 *Description*

21 Initializes a new instance of the  
22 **System.Web.UI.ConstructorNeedsTagAttribute** class.

23 ConstructorNeedsTagAttribute

24 *Example Syntax:*

25 ToString

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#]      public      ConstructorNeedsTagAttribute(bool      needsTag);  
[C++]     public:      ConstructorNeedsTagAttribute(bool      needsTag);  
[VB]      Public      Sub      New(ByVal      needsTag      As      Boolean)  
[JScript] public function ConstructorNeedsTagAttribute(needsTag : Boolean);
```

*Description*

Initializes a new instance of the **System.Web.UI.ConstructorNeedsTagAttribute** class. **true** to add a tag to a control; otherwise, **false**.

NeedsTag  
ToString

```
[C#]      public      bool      NeedsTag      {get;}  
[C++]     public:      __property      bool      get_NeedsTag();  
[VB]      Public      ReadOnly      Property      NeedsTag      As      Boolean  
[JScript] public function get NeedsTag() : Boolean;
```

*Description*

Indicates whether a control needs a tag in its constructor. This property is read-only.

TypeId  
Control class (System.Web.UI)  
ToString



*Description*

Defines the properties, methods, and events that are shared by all ASP.NET server controls.

This is the primary class that you derive from when you develop custom ASP.NET server controls.

Control

*Example Syntax:*

ToString

[C#]	public	Control();
[C++]	public:	Control();
[VB]	Public	Sub New()
[JScript]	public	function Control();

*Description*

Initializes a new instance of the **System.Web.UI.Control** class.

ChildControlsCreated

ToString

[C#]	protected	bool	ChildControlsCreated	{get; set;}
[C++]	protected:	__property	bool	get_ChildControlsCreated();protected:
	__property	void		set_ChildControlsCreated(bool);
[VB]	Protected	Property	ChildControlsCreated	As Boolean

```

1 [JScript] protected function get ChildControlsCreated() : Boolean;protected
2 function          set          ChildControlsCreated(Boolean);
3

```

#### 4 *Description*

5 Gets a value that indicates whether the server control's child controls have  
6 been created.

7 ClientID

8 ToString

9  
10 [C#] public virtual string ClientID {get;}

11 [C++] public: \_\_property virtual String\* get\_ClientID();

12 [VB] Overridable Public ReadOnly Property ClientID As String

13 [JScript] public function get ClientID() : String;

#### 14 15 *Description*

16 Gets the server control identifier generated by ASP.NET.

17 ASP.NET automatically generates a **ClientID** for a server control  
18 regardless of whether you have specified an **ID** property for it or not. A **ClientID**  
19 is assigned to all **System.Web.UI.LiteralControl** objects (text and HTML that is  
20 not processed on the server) on a page when it is requested. You can view the  
21 **ClientID** s of each control in a page's control hierarchy by enabling tracing for the  
22 page or application. The format for automatically generated **ClientID** s is Ctrl0 for  
23 the first control on a page, Ctrl1 for the next control, Ctrl2 for the third control,  
24 and so on until all controls for the page are given a **ClientID** value.

25 Context

## ToString

```
[C#]      protected      virtual      HttpContext      Context      {get;}
[C++]     protected:     __property virtual      HttpContext*      get_Context();
[VB]      Overridable Protected ReadOnly Property Context As HttpContext
[JScript] protected      function      get      Context()      :      HttpContext;
```

### *Description*

Gets the **System.Web.HttpContext** object associated with the server control for the current Web request.

This property gives you access to the **Context** object for the current Web request. The object provides properties that access the **Application** , **Session** , **Request** , **Response** , and other objects that contain information about the current HTTP request. It also provides methods that allow you to obtain configuration information and set or clear errors for the request.

### Controls

## ToString

```
[C#]      public      virtual      ControlCollection      Controls      {get;}
[C++]     public:     __property virtual      ControlCollection*      get_Controls();
[VB]      Overridable Public ReadOnly Property Controls As ControlCollection
[JScript] public      function      get      Controls()      :      ControlCollection;
```

### *Description*

1 Gets a **System.Web.UI.ControlCollection** object that represents the child  
2 controls for a specified server control in the UI hierarchy.

3 On an ASP.NET page, when controls are added declaratively between the  
4 opening and closing tags of a server control, ASP.NET automatically adds the  
5 controls to the containing server control's **System.Web.UI.ControlCollection** .  
6 Any HTML tags or text strings that are not processed on the server are compiled  
7 into **System.Web.UI.LiteralControl** objects. These are added to the collection  
8 like other server controls.

9 EnableViewState

10 ToString

11  
12 [C#] public virtual bool EnableViewState {get; set;}  
13 [C++] public: \_\_property virtual bool get\_EnableViewState();public: \_\_property  
14 virtual void set\_EnableViewState(bool);  
15 [VB] Overridable Public Property EnableViewState As Boolean  
16 [JScript] public function get EnableViewState() : Boolean;public function set  
17 EnableViewState(Boolean);  
18

19 *Description*

20 Gets or sets a value indicating whether the server control maintains its view  
21 state, and the view state of any child controls it contains, when the current page  
22 request ends.

23 You must enable view state for the server control to maintain its state  
24 across HTTP requests.

25 Events

## ToString

```
[C#]          protected          EventHandlerList          Events          {get;}
[C++]         protected:    __property    EventHandlerList*    get_Events();
[VB]   Protected   ReadOnly   Property   Events   As   EventHandlerList
[JScript]   protected   function   get   Events()   :   EventHandlerList;
```

### *Description*

Gets a list of event handler delegates for the control. This property is read-only.

This property is of type **System.ComponentModel.EventHandlerList** , which uses a linear search algorithm to find entries in the list of delegates. A linear search algorithm is inefficient when working with a large number of entries. Therefore, when you have a large list, finding entries with this property will be slow.

## HasChildViewState

## ToString

```
[C#]          protected          bool          HasChildViewState          {get;}
[C++]         protected:    __property    bool    get_HasChildViewState();
[VB]   Protected   ReadOnly   Property   HasChildViewState   As   Boolean
[JScript]   protected   function   get   HasChildViewState()   :   Boolean;
```

### *Description*

1 Gets a value indicating whether the current server control's child controls  
2 have any saved view-state settings.

3 You can avoid unnecessary calls to the  
4 **System.Web.UI.Control.ClearChildViewState** method by using this property to  
5 verify that any child controls of the server control are storing view-state  
6 information.

7 ID

8 ToString

9  
10 [C#] public virtual string ID {get; set;}

11 [C++] public: \_\_property virtual String\* get\_ID();public: \_\_property virtual void  
12 set\_ID(String\*);

13 [VB] Overridable Public Property ID As String

14 [JScript] public function get ID() : String;public function set ID(String);

15  
16 *Description*

17 Gets or sets the programmatic identifier assigned to the server control.

18 Setting this property on a server control provides you with programmatic  
19 access to the server control's properties, events, and methods. This property can be  
20 set by Web developers by declaring an **id** attribute in the opening tag of an  
21 ASP.NET server control.

22 IsTrackingViewState

23 ToString

24  
25 [C#] protected bool IsTrackingViewState {get;}

```

1 [C++]    protected:    __property    bool    get_IsTrackingViewState();
2 [VB]    Protected    ReadOnly    Property    IsTrackingViewState    As    Boolean
3 [JScript]    protected    function    get    IsTrackingViewState()    :    Boolean;

```

#### *Description*

Gets a value that indicates whether the server control is saving changes to its view state.

For a sample custom server control that uses this property, see .

NamingContainer

ToString

```

12 [C#]    public    virtual    Control    NamingContainer    {get;}
13 [C++]    public:    __property    virtual    Control*    get_NamingContainer();
14 [VB]    Overridable    Public    ReadOnly    Property    NamingContainer    As    Control
15 [JScript]    public    function    get    NamingContainer()    :    Control;

```

#### *Description*

Gets a reference to the server control's naming container, which creates a unique namespace for differentiating between server controls with the same **System.Web.UI.Control.ID** property value.

Each page in an ASP.NET Web application contains a hierarchy of controls. This hierarchy is not dependent on whether a control generates UI visible to the user. The naming container for a given control is the parent control above it in the hierarchy that implements the **INamingContainer** interface. A server

control that implements this interface creates a unique namespace for the **ID** property values of its child server controls.

Page

ToString

```
[C#]      public      virtual      Page      Page      {get;      set;}
```

```
[C++] public: __property virtual Page* get_Page();public: __property virtual void  
set_Page(Page*);
```

```
[VB]      Overridable      Public      Property      Page      As      Page
```

```
[JScript] public function get Page() : Page;public function set Page(Page);
```

### *Description*

Gets a reference to the **System.Web.UI.Page** instance that contains the server control.

This property reflects the name of the .aspx file associated with the current Web request. In the control hierarchy, name of the page is **\_PAGE** by default. It is assigned to a type defined by the name of the .aspx file.

Parent

ToString

```
[C#]      public      virtual      Control      Parent      {get;}
```

```
[C++] public:      __property      virtual      Control*      get_Parent();
```

```
[VB]      Overridable      Public      ReadOnly      Property      Parent      As      Control
```

```
[JScript] public function get Parent() : Control;
```



## Description

Gets a reference to the server control's parent control in the page UI hierarchy.

Whenever a page is requested, a hierarchy of server controls on that page is built. This property allows you to determine the parent control of the current server control in that hierarchy, and to program against it.

Site

ToString

```
[C#]      public      ISite      Site      {get;      set;}
```

```
[C++] public:  __property  ISite*  get_Site();public:  __property  void  
set_Site(ISite*);
```

```
[VB]      Public      Property      Site      As      ISite
```

```
[JScript] public function get Site() : ISite;public function set Site(ISite);
```

## Description

Gets information about the Web site to which the server control belongs.

Sites bind a **Component** to a **Container** and enable communication between them. They also provide a way for the container to manage its components.

TemplateSourceDirectory

ToString

```
[C#]      public      virtual      string      TemplateSourceDirectory      {get;}
```

```

1 [C++] public: __property virtual String* get_TemplateSourceDirectory();
2 [VB] Overridable Public ReadOnly Property TemplateSourceDirectory As String
3 [JScript] public function get TemplateSourceDirectory() : String;
4

```

#### *Description*

Gets the virtual directory of the **System.Web.UI.Page** or **System.Web.UI.UserControl** that contains the current server control.

UniqueID

ToString

```

11 [C#] public virtual string UniqueID {get;}
12 [C++] public: __property virtual String* get_UniqueID();
13 [VB] Overridable Public ReadOnly Property UniqueID As String
14 [JScript] public function get UniqueID() : String;
15

```

#### *Description*

Gets the unique, hierarchically-qualified identifier for the server control.

This property differs from the **ID** property, in that the **UniqueID** property includes the identifier for the server control's naming container. This identifier is generated automatically when a page request is processed.

ViewState

ToString

```

24 [C#] protected virtual StateBag ViewState {get;}
25 [C++] protected: __property virtual StateBag* get_ViewState();

```

```

1 [VB] Overridable Protected ReadOnly Property ViewState As StateBag
2 [JScript]    protected    function    get    ViewState()    :    StateBag;

```

#### *Description*

Gets a dictionary of state information that allows you to save and restore the view state of a server control across multiple requests for the same page.

A server control's view state is the accumulation of all its property values. In order to preserve these values across HTTP requests, ASP.NET server controls use this property, which is an instance of the **StateBag** class, to store the property values. The values are then passed as a variable to a hidden field when subsequent requests are processed. For more information about view state, see . For more information about saving server control view state, see .

ViewStateIgnoresCase

ToString

```

16 [C#]    protected    virtual    bool    ViewStateIgnoresCase    {get;}
17 [C++]    protected:    __property    virtual    bool    get_ViewStateIgnoresCase();
18 [VB] Overridable Protected ReadOnly Property ViewStateIgnoresCase As
19 Boolean
20 [JScript]    protected    function    get    ViewStateIgnoresCase()    :    Boolean;

```

#### *Description*

Gets a value that indicates whether the **System.Web.UI.StateBag** object is case-insensitive.

Visible

1 ToString

2

3 [C#] public virtual bool Visible {get; set;}

4 [C++] public: \_\_property virtual bool get\_Visible();public: \_\_property virtual void

5 set\_Visible(bool);

6 [VB] Overridable Public Property Visible As Boolean

7 [JScript] public function get Visible() : Boolean;public function set

8 Visible(Boolean);

9

10 *Description*

11 Gets or sets a value that indicates whether a server control is rendered as UI

12 on the page.

13 Server controls that are not visible take up no space on a Web Forms page

14 when it is displayed.

15 ToString

16

17 [C#] public event EventHandler DataBinding;

18 [C++] public: \_\_event EventHandler\* DataBinding;

19 [VB] Public Event DataBinding As EventHandler

20

21 *Description*

22 Occurs when the server control binds to a data source.

23 This event notifies the server control to perform any data binding logic that

24 has been written for it.

25 ToString

```

1
2 [C#]      public      event      EventHandler      Disposed;
3 [C++]     public:     __sealed   __event   EventHandler*   Disposed;
4 [VB]      NotOverridable   Public   Event   Disposed   As   EventHandler
5

```

### 6 *Description*

7 Occurs when a server control is released from memory, which is the last  
8 stage of the server control lifecycle when an ASP.NET page is requested.

9 Resources that require significant processor time, such as database  
10 connections, should be released with this event.

### 11 ToString

```

12
13 [C#]      public      event      EventHandler      Init;
14 [C++]     public:     __event      EventHandler*      Init;
15 [VB]      Public      Event      Init      As      EventHandler
16

```

### 17 *Description*

18 Occurs when the server control is initialized, which is the first step in the its  
19 lifecycle.

20 Server controls should perform any initialization steps that are required to  
21 create and set up an instance. You cannot use view-state information within this  
22 event; it is not populated yet. You should not access another server control during  
23 this event, regardless of whether it is a child or parent to this control. Other server  
24 controls are not certain to be created and ready for access.

### 25 ToString

[C#]	public	event	EventHandler	Load;
[C++]	public:	__event	EventHandler*	Load;
[VB]	Public	Event	Load As	EventHandler

*Description*

Occurs when the server control is loaded into the **System.Web.UI.Page** object.

Notifies the server control to perform any processing steps that are set to occur on each page request. You can use view state information with this event. You can also access other server controls within the page's control hierarchy.

*ToString*

[C#]	public	event	EventHandler	PreRender;
[C++]	public:	__event	EventHandler*	PreRender;
[VB]	Public	Event	PreRender As	EventHandler

*Description*

Occurs when the server control is about to render to its containing **System.Web.UI.Page** object.

Use this event to perform any updates before the output the server control is rendered to the page. Any changes in the view state of the server control can be saved during this event. Such changes made in the rendering phase will not be saved.

*ToString*

```

1
2 [C#]      public      event      EventHandler      Unload;
3 [C++]     public:    __event    EventHandler*      Unload;
4 [VB]      Public     Event      Unload      As      EventHandler
5

```

## 6 *Description*

7 Occurs when the server control is unloaded from memory.

8 Server controls must perform any final clean-up, such as closing files,  
9 database connections and discarding objects, during this stage of the control  
10 lifecycle before the instance is unloaded.

## 11 AddedControl

```

12
13 [C#] protected internal virtual void AddedControl(Control control, int index);
14 [C++] protected public: virtual void AddedControl(Control* control, int index);
15 [VB] Overridable Protected Friend Dim Sub AddedControl(ByVal control As
16 Control,          ByVal          index          As          Integer)
17 [JScript] package function AddedControl(control : Control, index : int);
18

```

## 19 *Description*

## 20 AddParsedSubObject

```

21
22 [C#]      protected      virtual      void      AddParsedSubObject(object      obj);
23 [C++]     protected:    virtual      void      AddParsedSubObject(Object*      obj);
24 [VB]      Overridable Protected Sub AddParsedSubObject(ByVal obj As Object)
25 [JScript] protected      function      AddParsedSubObject(obj      :      Object);

```

1  
2 *Description*

3       Notifies the server control that an element, either XML or HTML, was  
4       parsed, and adds the element to the server control's  
5       **System.Web.UI.ControlCollection** object.

6       Unless you override it, this method automatically adds  
7       **System.Web.UI.LiteralControl** objects to the server control's **ControlCollection**  
8       object. This collection is accessible through **System.Web.UI.Control.Controls**  
9       property. An **System.Object** that represents the parsed element.

10       BuildProfileTree

11  
12       [C#] protected void BuildProfileTree(string parentId, bool calcViewState);  
13       [C++] protected: void BuildProfileTree(String\* parentId, bool calcViewState);  
14       [VB] Protected Sub BuildProfileTree(ByVal parentId As String, ByVal  
15       calcViewState                               As                               Boolean)  
16       [JScript] protected function BuildProfileTree(parentId : String, calcViewState :  
17       Boolean);

18  
19 *Description*

20       Gathers information about the server control and delivers it to the  
21       **System.Web.UI.Page.Trace** property to be displayed when tracing is enabled for  
22       the page.

23       This property gathers the information necessary about a page's UI hierarchy  
24       and passes it to the page's **Trace** property. When you enable tracing, either for a  
25       page or for your application, this information is displayed in the **Control Tree**



section of the trace output. Trace output for a page is appended to the end of the page; while trace output for an application can be viewed from the trace viewer (trace.axd file) which is stored in the application's root directory. For more information about tracing, see . The identifier of the control's parent. A Boolean that indicates whether the view-state size is calculated.

## ClearChildViewState

[C#]	protected	void	ClearChildViewState();
[C++]	protected:	void	ClearChildViewState();
[VB]	Protected	Sub	ClearChildViewState()
[JScript]	protected	function	ClearChildViewState();

## Description

Deletes the view-state information for all of the server control's child controls.

This method is commonly used when you override the **System.Web.UI.Control.DataBind** method when developing templated data-bound server controls. If you do not call this method, child control view-state information can be written to a parent server control, only to be overridden when data binding occurs.

## CreateChildControls

[C#]	protected	virtual	void	CreateChildControls();
[C++]	protected:	virtual	void	CreateChildControls();
[VB]	Overridable	Protected	Sub	CreateChildControls()

[JScript]                      protected                      function                      CreateChildControls();

### *Description*

Notifies server controls that use composition-based implementation to create any child controls they contain in preparation for posting back or rendering.

When you develop a composite or templated server control, you must override this method. For more information, see and .

### CreateControlCollection

[C#]            protected            virtual            ControlCollection            CreateControlCollection();

[C++]            protected:            virtual            ControlCollection\*            CreateControlCollection();

[VB]            Overridable            Protected            Function            CreateControlCollection()            As  
ControlCollection

[JScript]            protected            function            CreateControlCollection()            :            ControlCollection;

### *Description*

Creates a new **System.Web.UI.ControlCollection** object to hold the child controls (both literal and server) of the server control.

*Return Value:* A **ControlCollection** object that contains the current server control's child server controls.

### DataBind

[C#]                      public                      virtual                      void                      DataBind();

[C++]                      public:                      virtual                      void                      DataBind();

[VB]                      Overridable                      Public                      Sub                      DataBind()

1 [JScript] public function DataBind();

3 *Description*

4 Binds a data source to the invoked server control and all of its child  
5 controls.

6 Use this method to bind data from a source to a server control. This method  
7 is commonly used after retrieving a data set through a database query.

8 **Dispose**

10 [C#] public virtual void Dispose();

11 [C++] public: virtual void Dispose();

12 [VB] Overridable Public Sub Dispose()

13 [JScript] public function Dispose();

15 *Description*

16 Enables a server control to perform final clean up before it is released from  
17 memory.

18 Call **Dispose** when you are finished using the **System.Web.UI.Control** .  
19 The **Dispose** method leaves the **Control** in an unusable state. After calling this  
20 method, you must release all references to the control so the memory it was  
21 occupying can be reclaimed by garbage collection.

22 **EnsureChildControls**

24 [C#] protected virtual void EnsureChildControls();

25 [C++] protected: virtual void EnsureChildControls();

[VB]	Overridable	Protected	Sub	EnsureChildControls()
[JScript]	protected	function		EnsureChildControls();

*Description*

Checks if the server control contains child controls. If it does not, it creates child controls.

If the server control contains nothing but literal content or HTML that is parsed and compiled as a **System.Web.UI.LiteralControl** object, it adds that object to the server control's **System.Web.UI.ControlCollection** object. The instance of the **LiteralControl** then becomes a child of the server control.

FindControl

[C#]	public	virtual	Control	FindControl(string id);
[C++]	public:	virtual	Control*	FindControl(String* id);
[VB]	Overridable	Public	Function	FindControl(ByVal id As String) As Control
[JScript]	public	function	FindControl(id : String) : Control;	Searches the current
	naming	container	for	the specified server control.

*Description*

Searches the current naming container for a server control with the specified *id* parameter.

*Return Value:* The specified control, or 0 if the specified control does not exist. The identifier for the control to be found.

FindControl

```

1
2 [C#] protected virtual Control FindControl(string id, int pathOffset);
3 [C++] protected: virtual Control* FindControl(String* id, int pathOffset);
4 [VB] Overridable Protected Function FindControl(ByVal id As String, ByVal
5 pathOffset As Integer) As Control
6 [JScript] protected function FindControl(id : String, pathOffset : int) : Control;
7

```

### Description

Searches the current naming container for a server control with the specified *id* and an integer, specified in the *pathOffset* parameter, that aids in the search.

*Return Value:* The specified control, or 0 if the specified control does not exist. The identifier for the control to be found. The number of controls up the page control hierarchy needed to reach a naming container.

### HasControls

```

15
16
17 [C#] public virtual bool HasControls();
18 [C++] public: virtual bool HasControls();
19 [VB] Overridable Public Function HasControls() As Boolean
20 [JScript] public function HasControls() : Boolean;
21

```

### Description

Determines if the server control contains any child controls.

*Return Value:* **true** if the control contains other controls; otherwise, **false**.

Since this method simply determines if any child controls exist, it can enhance performance by allowing you to avoid an unnecessary call to the **Count** property on the server control's **System.Web.UI.ControlCollection** object. The **ControlCollection** is available on the control through the **System.Web.UI.Control.Controls** property.

#### IsLiteralContent

[C#]	protected	bool	IsLiteralContent();
[C++]	protected:	bool	IsLiteralContent();
[VB]	Protected	Function	IsLiteralContent() As Boolean
[JScript]	protected	function	IsLiteralContent() : Boolean;

#### Description

Determines if the server control holds only literal content.

**Return Value:** **true** if the server control is comprised of literal content; otherwise **false**.

When this method returns **true**, the server control's collection holds a single literal control. The content is then passed to the requesting browser as HTML.

#### LoadViewState

[C#]	protected	virtual	void	LoadViewState(object	savedState);
[C++]	protected:	virtual	void	LoadViewState(Object*	savedState);
[VB]	Overridable	Protected	Sub	LoadViewState(ByVal	savedState As Object)
[JScript]	protected	function	LoadViewState(savedState	:	Object);

## Description

Restores view-state information from a previous page request that was saved by the **System.Web.UI.Control.SaveViewState** method.

Override this method when you need to customize how a custom server control restores its view state. For more information, see . An **System.Object** that represents the control state to be restored.

### MapPathSecure

```
[C#]      protected      string      MapPathSecure(string      virtualPath);
[C++]     protected:     String*      MapPathSecure(String*      virtualPath);
[VB] Protected Function MapPathSecure(ByVal virtualPath As String) As String
[JavaScript] protected function MapPathSecure(virtualPath : String) : String;
```

## Description

Retrieves a mapped physical file path relative to the source file, if the requesting server control has sufficient security permissions to read the mapped result.

*Return Value:* The physical path to the requested file.

This method can only be used by server controls that have permissions to read files and which are part of fully trusted .dll files, such as System.Web.dll. This helps prevent security breaches. A relative or root relative URL.

### OnBubbleEvent

```
[C#] protected virtual bool OnBubbleEvent(object source, EventArgs args);
```

```

1 [C++] protected: virtual bool OnBubbleEvent(Object* source, EventArgs* args);
2 [VB] Overridable Protected Function OnBubbleEvent(ByVal source As Object,
3 ByVal args As EventArgs) As Boolean
4 [JScript] protected function OnBubbleEvent(source : Object, args : EventArgs) :
5 Boolean;

```

### 7 *Description*

8 Determines whether the event for the server control is passed up the page's  
9 UI server control hierarchy.

10 *Return Value:* **true** if the event has been cancelled; otherwise, **false** . The default  
11 is **false** .

12 ASP.NET server controls such as the  
13 **System.Web.UI.WebControls.Repeater** ,  
14 **System.Web.UI.WebControls.DataList** and  
15 **System.Web.UI.WebControls.DataGrid** Web controls can contain child controls  
16 that raise events. For example, each row in a **DataGrid** control can contain one or  
17 more buttons created dynamically by templates. The source of the event. An  
18 **System.EventArgs** object that contains the event data.

### 19 *OnDataBinding*

```

20
21 [C#] protected virtual void OnDataBinding(EventArgs e);
22 [C++] protected: virtual void OnDataBinding(EventArgs* e);
23 [VB] Overridable Protected Sub OnDataBinding(ByVal e As EventArgs)
24 [JScript] protected function OnDataBinding(e : EventArgs);

```



## Description

Raises the **DataBinding** event.

This method notifies a server control to perform any data binding logic that is associated with it. An **EventArgs** object that contains the event data.

### OnInit

[C#]       protected       virtual       void       OnInit(EventArgs    e);

[C++]       protected:       virtual       void       OnInit(EventArgs\*   e);

[VB]   Overridable   Protected   Sub   OnInit(ByVal   e   As   EventArgs)

[JScript]       protected       function       OnInit(e       :       EventArgs);

## Description

Raises the **Init** event.

When notified by this method, server controls must perform any initialization steps that are required to create and set up an instance. In this stage of the server control's lifecycle, the control's view state has yet to be populated. Additionally, you can not access other server controls when this method is called either, regardless of whether it is a child or parent to this control. Other server controls are not certain to be created and ready for access. An **System.EventArgs** object that contains the event data.

### OnLoad

[C#]       protected       virtual       void       OnLoad(EventArgs    e);

[C++]       protected:       virtual       void       OnLoad(EventArgs\*   e);

```

1 [VB] Overridable Protected Sub OnLoad(ByVal e As EventArgs)
2 [JScript] protected function OnLoad(e : EventArgs);

```

#### *Description*

Raises the **Load** event.

This method notifies the server control that it should perform actions common to each HTTP request for the page it is associated with, such as setting up a database query. At this stage in the page lifecycle, server controls in the hierarchy are created and initialized, view state is restored, and form controls reflect client-side data. The **System.EventArgs** object that contains the event data.

#### *OnPreRender*

```

13 [C#] protected virtual void OnPreRender(EventArgs e);
14 [C++] protected: virtual void OnPreRender(EventArgs* e);
15 [VB] Overridable Protected Sub OnPreRender(ByVal e As EventArgs)
16 [JScript] protected function OnPreRender(e : EventArgs);

```

#### *Description*

Raises the **System.Web.UI.Control.PreRender** event, which notifies the server control that

This method notifies the server control to perform any necessary prerendering steps prior to saving view state and rendering content. An **System.EventArgs** object that contains the event data.

#### *OnUnload*



more buttons created dynamically by templates. The source of the event. An **System.EventArgs** object that contains the event data.

RemovedControl

[C#] protected internal virtual void RemovedControl(Control control);

[C++] protected public: virtual void RemovedControl(Control\* control);

[VB] Overridable Protected Friend Dim Sub RemovedControl(ByVal control As Control)

[JScript] package function RemovedControl(control : Control);

### Description

Render

[C#] protected virtual void Render(HtmlTextWriter writer);

[C++] protected: virtual void Render(HtmlTextWriter\* writer);

[VB] Overridable Protected Sub Render(ByVal writer As HtmlTextWriter)

[JScript] protected function Render(writer : HtmlTextWriter);

### Description

Sends server control content to a provided **System.Web.UI.HtmlTextWriter** object, which writes the content to be rendered on the client.

When developing custom server controls, you can override this method to generate content for an ASP.NET page. For more information, see . The **HtmlTextWriter** object that receives the server control content.

## RenderChildren

```
[C#]    protected    virtual    void    RenderChildren(HtmlTextWriter    writer);  
[C++]    protected:    virtual    void    RenderChildren(HtmlTextWriter*    writer);  
[VB]    Overridable    Protected    Sub    RenderChildren(ByVal    writer    As  
HtmlTextWriter)  
[JScript]    protected    function    RenderChildren(writer    :    HtmlTextWriter);
```

### *Description*

Outputs the content of a server control's children to a provided **System.Web.UI.HtmlTextWriter** object, which writes the content to be rendered on the client.

This method uses the delegate specified in the **System.Web.UI.Control.SetRenderMethodDelegate(System.Web.UI.Render Method)** method to render any Active Server Pages (ASP) code on the page. If no ASP code exists on the page, this method renders any child controls for the server control. The **HtmlTextWriter** object that receives the child controls.

## RenderControl

```
[C#]    public    void    RenderControl(HtmlTextWriter    writer);  
[C++]    public:    void    RenderControl(HtmlTextWriter*    writer);  
[VB]    Public    Sub    RenderControl(ByVal    writer    As    HtmlTextWriter)  
[JScript]    public    function    RenderControl(writer    :    HtmlTextWriter);
```

### *Description*

Outputs server control content to a provided **System.Web.UI.HtmlTextWriter** object, then checks if tracing is enabled for the containing page and retrieves trace information about the server control.

If a server control's **System.Web.UI.Control.Visible** property is set to **true**, this method checks to see if tracing is enabled for the page, and renders the server control content to the page. The **HtmlTextWriter** object that receives the control content.

#### ResolveUrl

```
[C#]      public      string      ResolveUrl(string      relativeUrl);
[C++]     public:     String*      ResolveUrl(String*      relativeUrl);
[VB]      Public Function ResolveUrl(ByVal relativeUrl As String) As String
[JScript] public function ResolveUrl(relativeUrl : String) : String;
```

#### *Description*

Resolves a relative URL to the absolute URL where the page or user control associated with this request resides.

*Return Value:* The absolute URL.

This method uses the **System.Web.UI.Control.TemplateSourceDirectory** property to resolve to the absolute URL. The returned URL is for client use, and contains the session cookie if appropriate. The relative URL associated with the HTTP request.

#### SaveViewState

```
[C#]      protected      virtual      object      SaveViewState();
```

```

1 [C++]      protected:      virtual      Object*      SaveViewState();
2 [VB]   Overridable   Protected   Function   SaveViewState()   As   Object
3 [JScript]   protected   function   SaveViewState()   :   Object;

```

#### *Description*

Saves any server control view-state changes that have occurred since the time the page was posted back to the server.

*Return Value:* Returns the server control's current view state. If there is no view state associated with the control, this method returns **null**.

View state is the accumulation of the values of a server control's properties. These values are automatically placed in the server control's **System.Web.UI.Control.ViewState** property, which is an instance of the **System.Web.UI.StateBag** class. This property's value is then persisted to a string object after the save state stage of the server control life cycle. For more information, see .

#### **SetRenderMethodDelegate**

```

18 [C#]   public   void   SetRenderMethodDelegate(RenderMethod   renderMethod);
19 [C++]   public:   void   SetRenderMethodDelegate(RenderMethod*   renderMethod);
20 [VB]   Public   Sub   SetRenderMethodDelegate(ByVal   renderMethod   As
21   RenderMethod)
22 [JScript]   public   function   SetRenderMethodDelegate(renderMethod   :
23   RenderMethod);

```

#### *Description*

1 Assigns an event handler delegate to render the server control and its  
2 content into its parent control.

3 This method is supplied for implementation purposes only; you should  
4 never call it directly. The information necessary to pass to the delegate so that it  
5 can render the server control.

6 **IParserAccessor.AddParsedSubObject**

7  
8 [C#] void IParserAccessor.AddParsedSubObject(object obj);

9 [C++] void IParserAccessor::AddParsedSubObject(Object\* obj);

10 [VB] Sub AddParsedSubObject(ByVal obj As Object) Implements  
11 IParserAccessor.AddParsedSubObject

12 [JScript] function IParserAccessor.AddParsedSubObject(obj : Object);

13 **TrackViewState**

14  
15 [C#] protected virtual void TrackViewState();

16 [C++] protected: virtual void TrackViewState();

17 [VB] Overridable Protected Sub TrackViewState()

18 [JScript] protected function TrackViewState();

19  
20 *Description*

21 Causes tracking of view-state changes to the server control so they can be  
22 stored in the server control's **System.Web.UI.StateBag** object. This object is  
23 accessible through the **System.Web.UI.Control.ViewState** property.

24 Invoke this method when you develop templated data-bound controls. This  
25 method alerts ASP.NET to monitor changes to a server control's view state, which





```

1
2 [C#]          public          Type          ControlType          {get;}
3 [C++]         public:         __property    Type*          get_ControlType();
4 [VB]   Public   ReadOnly   Property   ControlType   As   Type
5 [JScript]    public   function   get   ControlType()   :   Type;
6

```

### *Description*

Gets the type for the control to be created. This can be a class type, an interface type, a value type and so on.

FChildrenAsProperties

TrackViewState

```

13 [C#]          protected        bool          FChildrenAsProperties          {get;}
14 [C++]         protected:       __property    bool          get_FChildrenAsProperties();
15 [VB]   Protected   ReadOnly   Property   FChildrenAsProperties   As   Boolean
16 [JScript]    protected   function   get   FChildrenAsProperties()   :   Boolean;
17

```

### *Description*

Checks if there are children controls to be parsed.

**true** if there are children controls to be parsed, **false** otherwise.

FIsNonParserAccessor

TrackViewState

```

24 [C#]          protected        bool          FIsNonParserAccessor          {get;}
25 [C++]         protected:       __property    bool          get_FIsNonParserAccessor();

```

```

1 [VB] Protected ReadOnly Property FIsNonParserAccessor As Boolean
2 [JScript] protected function get FIsNonParserAccessor() : Boolean;

```

#### *Description*

Checks if the control implements the **System.Web.UI.IParseAccessor** interface.

This is a read only property. It is **true** if the controls implements the **System.Web.UI.IParseAccessor** interface, otherwise it returns **false** .

HasAspCode

TrackViewState

```

12 [C#]          public          bool          HasAspCode          {get;}

```

```

13 [C++]          public:          __property          bool          get_HasAspCode();

```

```

14 [VB] Public ReadOnly Property HasAspCode As Boolean

```

```

15 [JScript] public function get HasAspCode() : Boolean;

```

#### *Description*

Gets a value indicating whether the control contains any Active Server Pages (ASP) code.

ID

TrackViewState

```

23 [C#]          public          string          ID          {get;          set;}

```

```

24 [C++] public:  __property String* get_ID();public:  __property void

```

```

25 set_ID(String*);

```

1 [VB] Public Property ID As String  
 2 [JScript] public function get ID() : String;public function set ID(String);

3  
 4 *Description*

5 Gets or sets the ID property for the control to be built.

6 InDesigner

7 TrackViewState

8  
 9 [C#] protected bool InDesigner {get;}

10 [C++] protected: \_\_property bool get\_InDesigner();

11 [VB] Protected ReadOnly Property InDesigner As Boolean

12 [JScript] protected function get InDesigner() : Boolean;

13  
 14 *Description*

15 Allows to change the behavior of control builders.

16 NamingContainerType

17 TrackViewState

18  
 19 [C#] public Type NamingContainerType {get;}

20 [C++] public: \_\_property Type\* get\_NamingContainerType();

21 [VB] Public ReadOnly Property NamingContainerType As Type

22 [JScript] public function get NamingContainerType() : Type;

23  
 24 *Description*

25 Gets the type assigned to the control's naming container.

```

1      Parser
2      ViewState
3
4  [C#]      protected      TemplateParser      Parser      {get;}
5  [C++]     protected:    __property      TemplateParser*      get_Parser();
6  [VB]      Protected      ReadOnly      Property      Parser      As      TemplateParser
7  [JScript]  protected      function      get      Parser()      :      TemplateParser;
8

```

#### *Description*

Gets the control builder parser.

TagName

ViewState

```

14 [C#]      public      string      TagName      {get;}
15 [C++]     public:    __property      String*      get_TagName();
16 [VB]      Public      ReadOnly      Property      TagName      As      String
17 [JScript]  public      function      get      TagName()      :      String;
18

```

#### *Description*

Gets the tag name for the control to be built.

AllowWhitespaceLiterals

```

23 [C#]      public      virtual      bool      AllowWhitespaceLiterals();
24 [C++]     public:    virtual      bool      AllowWhitespaceLiterals();
25 [VB]      Overridable Public Function AllowWhitespaceLiterals() As Boolean

```

1 [JScript] public function AllowWhitespaceLiterals() : Boolean;

3 *Description*

4 Allows spaces within a control to be created as a LiteralControl object. If  
5 you create a custom control builder, you can override this method

6 *Return Value:* **true** in all cases.

7 AppendLiteralString

9 [C#] public virtual void AppendLiteralString(string s);

10 [C++] public: virtual void AppendLiteralString(String\* s);

11 [VB] Overridable Public Sub AppendLiteralString(ByVal s As String)

12 [JScript] public function AppendLiteralString(s : String);

14 *Description*

15 Adds literal content to a control. Literal content is any text that is passed  
16 through to the browser without being processed on the server. For example, any  
17 HTML elements and text between their opening and closing tags are literal  
18 content. The content to add to the control.

19 AppendSubBuilder

21 [C#] public virtual void AppendSubBuilder(ControlBuilder subBuilder);

22 [C++] public: virtual void AppendSubBuilder(ControlBuilder\* subBuilder);

23 [VB] Overridable Public Sub AppendSubBuilder(ByVal subBuilder As  
24 ControlBuilder)

25 [JScript] public function AppendSubBuilder(subBuilder : ControlBuilder);

*Description*

Adds builders to the **System.Web.UI.ControlBuilder** object for any child controls that belong to the control. The **ControlBuilder** object assigned to the child control.

**CloseControl**

[C#]	public	virtual	void	CloseControl();
[C++]	public:	virtual	void	CloseControl();
[VB]	Overridable	Public	Sub	CloseControl()
[JScript]	public		function	CloseControl();

*Description*

Inserts the closing tag for the control.

**CreateBuilderFromType**

[C#]	public static ControlBuilder CreateBuilderFromType(TemplateParser parser, ControlBuilder parentBuilder, Type type, string tagName, string id, IDictionary attribs, int line, string sourceFileName);
[C++]	public: static ControlBuilder* CreateBuilderFromType(TemplateParser* parser, ControlBuilder* parentBuilder, Type* type, String* tagName, String* id, IDictionary* attribs, int line, String* sourceFileName);
[VB]	Public Shared Function CreateBuilderFromType(ByVal parser As TemplateParser, ByVal parentBuilder As ControlBuilder, ByVal type As Type, ByVal tagName As String, ByVal id As String, ByVal attribs As IDictionary,

```

1 ByVal line As Integer, ByVal sourceFileName As String) As ControlBuilder
2 [JScript] public static function CreateBuilderFromType(parser : TemplateParser,
3 parentBuilder : ControlBuilder, type : Type, tagName : String, id : String, attribs :
4 IDictionary, line : int, sourceFileName : String) : ControlBuilder;

```

## Description

Creates a **System.Web.UI.ControlBuilder** object for the specified tag.

*Return Value:* The builder that is responsible for creating the control. The **System.Web.UI.TemplateParser** object responsible for parsing the control. The **System.Web.UI.ControlBuilder** object responsible for building the control. The type assigned to the control that the builder will create. The name of the tag to be built. This allows the builder to support multiple tag types. The ID attribute assigned to the control. The **System.Collections.IDictionary** object that holds all of the specified tags attributes. The source file line number for the specified control. The name of the source file from which the control is to be created.

## GetChildControlType

```

18 [C#] public virtual Type GetChildControlType(string tagName, IDictionary
19 attribs);
20 [C++] public: virtual Type* GetChildControlType(String* tagName, IDictionary*
21 attribs);
22 [VB] Overridable Public Function GetChildControlType(ByVal tagName As
23 String, ByVal attribs As IDictionary) As Type
24 [JScript] public function GetChildControlType(tagName : String, attribs :
25 IDictionary) : Type;

```



## Description

Obtains the Type for the current control's children.

**Return Value:** If a default property exists, this method returns the default property builder. If the type is a collection, a collection builder; if the type is a template, a template builder. If none of the above, this method returns the type of the property, or **null** if there are no properties. The name of the current control's tag. An array of attributes contained in the current control.

## HasBody

[C#]            public            virtual            bool            HasBody();

[C++]           public:            virtual            bool            HasBody();

[VB]    Overridable    Public    Function    HasBody()    As    Boolean

[JScript]       public        function        HasBody()       :        Boolean;

## Description

Determines if a control has both an opening and closing tag.

**Return Value:** **true** if the control has an opening and closing tag; otherwise, **false**.

## HtmlDecodeLiterals

[C#]            public            virtual            bool            HtmlDecodeLiterals();

[C++]           public:            virtual            bool            HtmlDecodeLiterals();

[VB]    Overridable    Public    Function    HtmlDecodeLiterals()    As    Boolean

[JScript]       public        function        HtmlDecodeLiterals()       :        Boolean;

## Description

Checks if the literal string of an HTML control must be decoded.  
*Return Value:* **true** if the HTML control literal string is to be decoded, **false** otherwise.

## Init

```
[C#] public virtual void Init(TemplateParser parser, ControlBuilder parentBuilder,  
Type type, string tagName, string id, IDictionary attribs);  
[C++] public: virtual void Init(TemplateParser* parser, ControlBuilder*  
parentBuilder, Type* type, String* tagName, String* id, IDictionary* attribs);  
[VB] Overridable Public Sub Init(ByVal parser As TemplateParser, ByVal  
parentBuilder As ControlBuilder, ByVal type As Type, ByVal tagName As String,  
ByVal id As String, ByVal attribs As IDictionary)  
[JScript] public function Init(parser : TemplateParser, parentBuilder :  
ControlBuilder, type : Type, tagName : String, id : String, attribs : IDictionary);
```

## Description

Initializes the control builder when a Web request is made. The **System.Web.UI.TemplateParser** object responsible for parsing the control. The **System.Web.UI.ControlBuilder** object responsible for building the control. The type assigned to the control that the builder will create. The name of the tag to be built. This allows the builder to support multiple tag types. The ID attribute assigned to the control. The **System.Collections.IDictionary** object that holds all of the specified tags attributes.

1 NeedsTagInnerText

2

3 [C#] public virtual bool NeedsTagInnerText();

4 [C++] public: virtual bool NeedsTagInnerText();

5 [VB] Overridable Public Function NeedsTagInnerText() As Boolean

6 [JScript] public function NeedsTagInnerText() : Boolean;

7

8 *Description*

9 Determines if a control tag needs inner text. If so, the

10 **System.Web.UI.ControlBuilder.SetTagInnerText(System.String)** method must

11 be called.

12 *Return Value:* **true** if the control tag does not have any inner text; otherwise, **false**

13 . The default is **false** .

14 OnAppendToParentBuilder

15

16 [C#] public virtual void OnAppendToParentBuilder(ControlBuilder

17 parentBuilder);

18 [C++] public: virtual void OnAppendToParentBuilder(ControlBuilder\*

19 parentBuilder);

20 [VB] Overridable Public Sub OnAppendToParentBuilder(ByVal parentBuilder As

21 ControlBuilder)

22 [JScript] public function OnAppendToParentBuilder(parentBuilder :

23 ControlBuilder);

24

25 *Description*

1       Notifies the current control builder that it is being added to a parent control  
2 builder. The ControlBuilder object to which the current builder is added.

### 3       SetTagInnerText

4  
5 [C#]     public     virtual     void     SetTagInnerText(string     text);  
6 [C++]     public:     virtual     void     SetTagInnerText(String\*     text);  
7 [VB]   Overridable   Public   Sub   SetTagInnerText(ByVal   text   As   String)  
8 [JScript]     public     function     SetTagInnerText(text     :     String);  
9

### 10     *Description*

11       Provides the builder with the inner text of the control tag. The text to be  
12 provided.

13       ControlBuilderAttribute class (System.Web.UI)

### 14       ToString

### 17     *Description*

18       Specifies a custom **System.Web.UI.ControlBuilder** object for building a  
19 control within the ASP.NET parser. This class cannot be inherited.

20       This attribute prefixes the class to be used by the control builder at to create  
21 a custom control at run time.

### 22       ToString

23  
24 [C#]     public     static     readonly     ControlBuilderAttribute     Default;  
25 [C++]     public:     static     ControlBuilderAttribute\*     Default;

```

1 [VB] Public Shared ReadOnly Default As ControlBuilderAttribute
2 [JScript] public static var Default : ControlBuilderAttribute;
3

```

#### 4 *Description*

5 Specifies the new **System.Web.UI.ControlBuilderAttribute** object. By  
6 default, the new object is set to null. This field is read-only.

7 **ControlBuilderAttribute**

8 *Example Syntax:*

9 **ToString**

```

10
11 [C#] public ControlBuilderAttribute(Type builderType);

```

```

12 [C++] public: ControlBuilderAttribute(Type* builderType);

```

```

13 [VB] Public Sub New(ByVal builderType As Type)

```

```

14 [JScript] public function ControlBuilderAttribute(builderType : Type);
15

```

#### 16 *Description*

17 Specifies the control builder for a custom control.

18 The association of the control builder type with a custom control is  
19 obtained by preeceding the custom control builder class with the following  
20 attribute definition: [ControlBuilderAttribute(typeof(controlbuildertype))] control  
21 builder type

22 **BuilderType**

23 **ToString**

```

24
25 [C#] public Type BuilderType {get;}

```

1 [C++] public: \_\_property Type\* get\_BuilderType();  
 2 [VB] Public ReadOnly Property BuilderType As Type  
 3 [JScript] public function get BuilderType() : Type;

4  
 5 *Description*

6 Gets the type assigned to the control. This property is read-only.

7 TypeId

8 Equals

9  
 10 [C#] public override bool Equals(object obj);  
 11 [C++] public: bool Equals(Object\* obj);  
 12 [VB] Overrides Public Function Equals(ByVal obj As Object) As Boolean  
 13 [JScript] public override function Equals(obj : Object) : Boolean;

14  
 15 *Description*

16 Determines the control builder type for the specified control.

17 GetHashCode

18  
 19 [C#] public override int GetHashCode();  
 20 [C++] public: int GetHashCode();  
 21 [VB] Overrides Public Function GetHashCode() As Integer  
 22 [JScript] public override function GetHashCode() : int;

23  
 24 *Description*

25

Returns the hash code for the control object.

*Return Value:* **int** representing the control object hash code

IsDefaultAttribute

[C#] public override bool IsDefaultAttribute();

[C++] public: bool IsDefaultAttribute();

[VB] Overrides Public Function IsDefaultAttribute() As Boolean

[JScript] public override function IsDefaultAttribute() : Boolean;

#### *Description*

Determines the current is the default control builder.

*Return Value:* **true** if the current is the default control builder.

ControlCollection class (System.Web.UI)

ToString

#### *Description*

Provides a collection container that enables ASP.NET server controls to maintain a list of their child controls.

You can access any of the properties and methods of the **ControlCollection** class through the **System.Web.UI.Control.Controls** property. Since the **Control** class is the base class for all ASP.NET server controls, all server controls inherit this property.

ControlCollection

*Example Syntax:*

```

1      ToString
2
3  [C#]          public          ControlCollection(Control          owner);
4  [C++]          public:          ControlCollection(Control*          owner);
5  [VB]      Public      Sub      New(ByVal      owner      As      Control)
6  [JScript]      public      function      ControlCollection(owner      :      Control);
7

```

#### 8 *Description*

9        Initializes a new instance of the **System.Web.UI.ControlCollection** class  
10        for the parent server control specified in the *owner* parameter. The ASP.NET  
11        server control that the control collection is created for.

12        Count

13        ToString

```

14
15  [C#]          public          int          Count          {get;}
16  [C++]          public:          __property          int          get_Count();
17  [VB]      Public      ReadOnly      Property      Count      As      Integer
18  [JScript]      public      function      get      Count()      :      int;
19

```

#### 20 *Description*

21        Gets the number of server controls in the **ControlCollection** object for the  
22        specified ASP.NET server control.

23        IsReadOnly

24        ToString

25



```

1
2 [C#]          public          bool          IsReadOnly          {get;}
3 [C++]         public:         __property     bool          get_IsReadOnly();
4 [VB]   Public   ReadOnly   Property   IsReadOnly   As   Boolean
5 [JScript]   public   function   get   IsReadOnly()   :   Boolean;
6

```

#### *Description*

Gets a value indicating whether the **ControlCollection** object is read-only.

IsSynchronized

ToString

```

11
12 [C#]          public          bool          IsSynchronized          {get;}
13 [C++]         public:         __property     bool          get_IsSynchronized();
14 [VB]   Public   ReadOnly   Property   IsSynchronized   As   Boolean
15 [JScript]   public   function   get   IsSynchronized()   :   Boolean;
16

```

#### *Description*

Gets a value indicating whether the **ControlCollection** is synchronized.

Item

ToString

```

21
22 [C#]   public   virtual   Control   this[int   index]   {get;}
23 [C++]   public:   __property   virtual   Control*   get_Item(int   index);
24 [VB]   Overridable   Public   Default   ReadOnly   Property   Item(ByVal   index   As
25 Integer)                                     As   Control

```

1 [JScript]        returnValue        =        ControlCollectionObject.Item(index);

2

3 *Description*

4        Gets a reference to the server control at the specified index location in the  
5 **ControlCollection** object. The location of the server control in the  
6 **ControlCollection** .

7        Owner

8        ToString

9

10 [C#]            protected            Control            Owner            {get;}

11 [C++]           protected:           \_\_property           Control\*           get\_Owner();

12 [VB]    Protected    ReadOnly    Property    Owner    As    Control

13 [JScript]    protected    function    get    Owner()    :    Control;

14

15 *Description*

16        Gets the ASP.NET server control to which the  
17 **System.Web.UI.ControlCollection** object belongs.

18        SyncRoot

19        ToString

20

21 [C#]            public            object            SyncRoot            {get;}

22 [C++]           public:           \_\_property           Object\*           get\_SyncRoot();

23 [VB]    Public    ReadOnly    Property    SyncRoot    As    Object

24 [JScript]    public    function    get    SyncRoot()    :    Object;

25

## Description

Gets an object that can be used to synchronize access to the collection of controls.

### Add

[C#]        public        virtual        void        Add(Control        child);

[C++]        public:        virtual        void        Add(Control\*        child);

[VB]    Overridable    Public    Sub    Add(ByVal    child    As    Control)

[JScript]        public        function        Add(child        :        Control);

## Description

Adds the specified **System.Web.UI.Control** object to the collection.

The new control is added to the end of an ordinal index array. The control can be an instance of any ASP.NET server control, a custom server control you create, or a literal control. The **Control** object to add to the collection.

### AddAt

[C#]    public    virtual    void    AddAt(int    index,    Control    child);

[C++]    public:    virtual    void    AddAt(int    index,    Control\*    child);

[VB] Overridable Public Sub AddAt(ByVal index As Integer, ByVal child As Control)

[JScript]    public    function    AddAt(index    :    int,    child    :    Control);

## Description

1 Adds the specified **System.Web.UI.Control** object to the collection at the  
2 specified index location.

3 The added control can be an instance of any ASP.NET server control, a  
4 custom server control you create, or a literal control. The location in the array to  
5 add the child control. The **Control** object to add to the collection.

#### 6 Clear

8 [C#]	public	virtual	void	Clear();
9 [C++]	public:	virtual	void	Clear();
10 [VB]	Overridable	Public	Sub	Clear()
11 [JScript]	public	function		Clear();

#### 13 Description

14 Removes all controls from the current server control's  
15 **System.Web.UI.ControlCollection** object.

16 Use this method to empty a custom control's **ControlCollection** when you  
17 override the **System.Web.UI.Control.CreateChildControls** and  
18 **System.Web.UI.Control.DataBind** methods. Do this when you develop  
19 composite, templated controls or templated data bound controls. For more  
20 information, see .

#### 21 Contains

23 [C#]	public	virtual	bool	Contains(Control c);
24 [C++]	public:	virtual	bool	Contains(Control* c);
25 [VB]	Overridable	Public	Function	Contains(ByVal c As Control) As Boolean

1 [JScript] public function Contains(c : Control) : Boolean;

3 *Description*

4 Determines whether the specified server control is in the parent server  
5 control's **System.Web.UI.ControlCollection** object.

6 *Return Value:* **true** if the specified server control exists in the collection;  
7 otherwise, **false** . The server control to search for in the collection.

8 CopyTo

10 [C#] public void CopyTo(Array array, int index);

11 [C++] public: \_\_sealed void CopyTo(Array\* array, int index);

12 [VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As  
13 Integer)

14 [JScript] public function CopyTo(array : Array, index : int);

16 *Description*

17 Copies the child controls stored in the **System.Web.UI.ControlCollection**  
18 object to an **System.Array** object, beginning at the specified index location in the  
19 **Array** . The **Array** to copy the child controls to. The zero-based relative index in  
20 *array* where copying begins.

21 GetEnumerator

23 [C#] public IEnumerator GetEnumerator();

24 [C++] public: \_\_sealed IEnumerator\* GetEnumerator();

25 [VB] NotOverridable Public Function GetEnumerator() As IEnumerator

1 [JScript] public function GetEnumerator() : IEnumerator;

3 *Description*

4 Retrieves an enumerator that can iterate through the **ControlCollection**  
5 object.

6 *Return Value:* The enumerator to iterate through the collection.

7 **IndexOf**

9 [C#] public virtual int IndexOf(Control value);

10 [C++] public: virtual int IndexOf(Control\* value);

11 [VB] Overridable Public Function IndexOf(ByVal value As Control) As Integer

12 [JScript] public function IndexOf(value : Control) : int;

14 *Description*

15 Retrieves the index of a specified **System.Web.UI.Control** object in the  
16 collection.

17 *Return Value:* The index of the specified server control. If the server control is not  
18 currently a member of the collection, it returns -1.

19 Use this method to determine the index location of the specified server  
20 control in the **ControlCollection** object. The **Control** object for which the index  
21 is returned.

22 **Remove**

24 [C#] public virtual void Remove(Control value);

25 [C++] public: virtual void Remove(Control\* value);

```

1 [VB] Overridable Public Sub Remove(ByVal value As Control)
2 [JScript]      public      function      Remove(value      :      Control);

```

#### *Description*

Removes the specified server control from the parent server control's **System.Web.UI.ControlCollection** object.

To remove a control from an index location, use the **System.Web.UI.ControlCollection.RemoveAt(System.Int32)** method. The server control to be removed.

#### **RemoveAt**

```

12 [C#]      public      virtual      void      RemoveAt(int      index);
13 [C++]      public:      virtual      void      RemoveAt(int      index);
14 [VB] Overridable Public Sub RemoveAt(ByVal index As Integer)
15 [JScript]      public      function      RemoveAt(index      :      int);

```

#### *Description*

Removes a child control, at the specified index location, from the **System.Web.UI.ControlCollection** object.

To remove a control from the collection based on its value, use the **System.Web.UI.ControlCollection.Remove(System.Web.UI.Control)** method. The ordinal index of the server control to be removed from the collection.

CssStyleCollection class (System.Web.UI)

ToString

### Description

Contains the HTML cascading-style sheets (CSS) inline style attributes for a specified HTML server control. This class cannot be inherited.

Any style declared for a particular HTML server control is added to the collection when the containing Web Forms page is parsed. It automatically parses and exposes CSS properties through a dictionary pattern API. You can manipulate any CSS property on a server control through the **System.Web.UI.HtmlControls.HtmlControl.Style** property. Simply use the CSS property's key and value in the indexed collection.

Count

ToString

[C#]	public	int	Count	{get;}
[C++]	public:	__property	int	get_Count();
[VB]	Public	ReadOnly Property	Count	As Integer
[JScript]	public	function	get Count()	: int;

### Description

Gets the number of items in the **CssStyleCollection** object.

Item

ToString

[C#]	public	string	this[string key]	{get; set;}
------	--------	--------	------------------	-------------



```

1 [C++] public: __property String* get_Item(String* key);public: __property void
2 set_Item(String* key, String*);
3 [VB] Public Default Property Item(ByVal key As String) As String
4 [JScript] returnValue =
5 CssStyleCollectionObject.Item(key);CssStyleCollectionObject.Item(key) =
6 returnValue;
7

```

### Description

Gets or sets a specified CSS value for the specified HTML server control.  
The index to the CSS attribute.

Keys

ToString

```

14 [C#] public ICollection Keys {get;}
15 [C++] public: __property ICollection* get_Keys();
16 [VB] Public ReadOnly Property Keys As ICollection
17 [JScript] public function get Keys() : ICollection;
18

```

### Description

Gets a collection of keys to all the styles in the **CssStyleCollection** object for a specific HTML server control.

Add

```

24 [C#] public void Add(string key, string value);
25 [C++] public: void Add(String* key, String* value);

```

```

1 [VB] Public Sub Add(ByVal key As String, ByVal value As String)
2 [JScript] public function Add(key : String, value : String);

```

#### Description

Adds a style item to the **CssStyleCollection** object. The index assigned to the new style in the collection. The style to store in the collection.

#### Clear

[C#]	public	void	Clear();
[C++]	public:	void	Clear();
[VB]	Public	Sub	Clear()
[JScript]	public	function	Clear();

#### Description

Removes all style items from the **CssStyleCollection** object.

#### Remove

[C#]	public	void	Remove(string	key);
[C++]	public:	void	Remove(String*	key);
[VB]	Public	Sub	Remove(ByVal	key As String)
[JScript]	public	function	Remove(key	: String);

#### Description

Removes a style item from the **CssStyleCollection** object. The index of the style item to remove.

1

2

3

4

7

8

9

10

13

14

5

16

19

21

22

23

1 As String) As Object  
 2 [JScript] public static function Eval(container : Object, expression : String) :  
 3 Object; Uses reflection to parse and evaluate a data-binding expression against an  
 4 object at runtime. This method allows RAD designers, such as Visual Studio.NET,  
 5 to easily generate and parse data-binding syntax. This method can also be used  
 6 declaratively on a Web Forms page to simplify casting to text that can be  
 7 displayed in a browser.

### 9 *Description*

10 Evaluates data binding expressions at runtime.

11 *Return Value:* An **System.Object** that results from the evaluation of the data-  
 12 binding expression.

13 While this method is automatically called when you create data bindings in  
 14 a RAD designer, you can also use it declaratively if you want to simplify the  
 15 casting to a text string to be displayed on a browser. To do so, you must place the  
 16 tags, which are also used in standard ASP.NET data binding, around the data-  
 17 binding expression. The object reference against which the expression is  
 18 evaluated. This must be a valid object identifier in the page's specified language.  
 19 The navigation path from the *container* to the property value to be placed in the  
 20 bound control property. This must be a string type of property or field names  
 21 separated by dots, such as **Table.DefaultView.Price** .

22 Eval

24 [C#] public static string Eval(object container, string expression, string format);

25 [C++] public: static String\* Eval(Object\* container, String\* expression, String\*

format);

[VB] Public Shared Function Eval(ByVal container As Object, ByVal expression

As String, ByVal format As String) As String

[JScript] public static function Eval(container : Object, expression : String, format

: String) : String;

### *Description*

Evaluates data binding expressions at runtime and formats the output as text to be displayed in the requesting browser.

*Return Value:* A **System.String** that results from the evaluation of the data binding expression and conversion to a string type. This will be displayed by the requesting browser.

To learn more about format strings in the .NET Framework, see . The object reference against which the expression is evaluated. This must be a valid object identifier in the page's specified language. The navigation path from the *container* to the property value to be placed in the bound control property. This must be a string of property or field names separated by dots, such as **Table.DefaultView.Price** . A .NET Framework format string that converts the **System.Object** (which results from the evaluation of the data binding expression) to a **System.String** that can be displayed by the requesting browser.

### **GetIndexedPropertyValue**

[C#] public static object GetIndexedPropertyValue(object container, string expr);

[C++] public: static Object\* GetIndexedPropertyValue(Object\* container, String\* expr);

```

1 [VB] Public Shared Function GetIndexedPropertyValue(ByVal container As
2 Object,      ByVal      expr      As      String)      As      Object
3 [JScript] public static function GetIndexedPropertyValue(container : Object, expr :
4 String)      :      Object;

```

#### *Description*

##### GetIndexedPropertyValue

```

9 [C#] public static string GetIndexedPropertyValue(object container, string
10 propName,      string      format);

```

```

11 [C++] public: static String* GetIndexedPropertyValue(Object* container, String*
12 propName,      String*      format);

```

```

13 [VB] Public Shared Function GetIndexedPropertyValue(ByVal container As
14 Object, ByVal propName As String, ByVal format As String) As String

```

```

15 [JScript] public static function GetIndexedPropertyValue(container : Object,
16 propName      :      String,      format      :      String)      :      String;

```

#### *Description*

##### GetPropertyValue

```

21 [C#] public static object GetPropertyValue(object container, string propName);

```

```

22 [C++] public: static Object* GetPropertyValue(Object* container, String*
23 propName);

```

```

24 [VB] Public Shared Function GetPropertyValue(ByVal container As Object,
25 ByVal      propName      As      String)      As      Object

```

1 [JScript] public static function GetPropertyValue(container : Object, propName :  
2 String) : Object;

3  
4 *Description*

5     GetPropertyValue

6  
7 [C#] public static string GetPropertyValue(object container, string propName,  
8 string format);

9 [C++] public: static String\* GetPropertyValue(Object\* container, String\*  
10 propName, String\* format);

11 [VB] Public Shared Function GetPropertyValue(ByVal container As Object,  
12 ByVal propName As String, ByVal format As String) As String

13 [JScript] public static function GetPropertyValue(container : Object, propName :  
14 String, format : String) : String;

15  
16 *Description*

17     DataBinding class (System.Web.UI)

18     ToString

19  
20  
21 *Description*

22     Contains information about a single data-binding expression in an  
23 ASP.NET server control, which allows rapid-application development (RAD)  
24 designers, such as Visual Studio .NET, to create data-binding expressions at  
25 design time. This class cannot be inherited.

Each data-binding expression in a server control is represented at design time by an instance of the **DataBinding** class. Any server control that contains one or more data-binding expressions has a **System.Web.UI.DataBindingCollection** object that contains the **DataBinding** objects. This collection is accessible by implementing the **System.Web.UI.IDataBindingsAccessor** interface. Any **DataBinding** or **DataBindingCollection** objects associated with a server control exist only at design time. They do not exist at runtime and, therefore, are not accessible then.

**DataBinding**

*Example Syntax:*

**ToString**

[C#] public DataBinding(string propertyName, Type propertyType, string expression);

[C++] public: DataBinding(String\* propertyName, Type\* propertyType, String\* expression);

[VB] Public Sub New(ByVal propertyName As String, ByVal propertyType As Type, ByVal expression As String)

[JScript] public function DataBinding(propertyName : String, propertyType : Type, expression : String);

### *Description*

Initializes a new instance of the **System.Web.UI.DataBinding** class. The property to data bind to. The .NET Framework type of the property to data bind to. The data-binding expression to be evaluated.



Expression

ToString

[C#]            public            string            Expression            {get;            set;}

[C++] public: \_\_property String\* get\_Expression();public: \_\_property void  
set\_Expression(String\*);

[VB]            Public            Property            Expression            As            String

[JScript] public function get Expression() : String;public function set  
Expression(String);

#### *Description*

Gets or sets the data-binding expression to be evaluated.

Whether or not you bind data to a server control property, data-binding  
expressions take the following form: .

PropertyName

ToString

[C#]            public            string            PropertyName            {get;}

[C++] public:            \_\_property            String\*            get\_PropertyName();

[VB]            Public            ReadOnly            Property            PropertyName            As            String

[JScript]            public            function            get            PropertyName()            :            String;

#### *Description*

Gets the name of the ASP.NET server control property to be data bound  
against.

```

1      PropertyType
2      ToString
3
4  [C#]      public      Type      PropertyType      {get;}
5  [C++]      public:      __property      Type*      get_PropertyType();
6  [VB]      Public      ReadOnly      Property      PropertyType      As      Type
7  [JScript]      public      function      get      PropertyType()      :      Type;
8

```

### *Description*

Gets the .NET Framework type of the data-bound ASP.NET server control property.

### *Equals*

```

14 [C#]      public      override      bool      Equals(object      obj);
15 [C++]      public:      bool      Equals(Object*      obj);
16 [VB]      Overrides Public Function Equals(ByVal obj As Object) As Boolean
17 [JScript]      public      override      function      Equals(obj : Object) : Boolean;
18

```

### *Description*

Determines whether the specified object is the same instance of the **System.Web.UI.DataBinding** class as the current object.

*Return Value:* **true** if the data binding property names match; otherwise, **false** .

The object to compare against the current **DataBinding** object.

### *GetHashCode*

```

1
2 [C#]          public          override          int          GetHashCode();
3 [C++]          public:          int          GetHashCode();
4 [VB]  Overrides  Public  Function  GetHashCode()  As  Integer
5 [JScript]  public  override  function  GetHashCode()  :  int;
6

```

### *Description*

Retrieves the hash code for an instance of the **System.Web.UI.DataBinding** object.

*Return Value:* A 32-bit signed integer hash code.

**DataBinding** objects are placed in a **System.Web.UI.DataBindingCollection** object, accessible at design time through the server control's **System.Web.UI.Control.DataBindings** property, which is a hash table that represents the bindings on a control. There can only be one binding per property, so the hashcode computation should match the **System.Web.UI.DataBinding.Equals(System.Object)** implementation and only take the property name into account.

**DataBindingCollection** class (System.Web.UI)

**ToString**

### *Description*

Provides a collection of **System.Web.UI.DataBinding** objects for an ASP.NET server control. This class cannot be inherited.

This collection is a hash table containing all of the **System.Web.UI.DataBinding** objects on an ASP.NET server control. You can access the objects contained in this collection by implementing the **System.Web.UI.IDataBindingsAccessor** interface. Any **DataBinding** or **DataBindingCollection** objects associated with a server control exist only at design time. They do not exist at runtime and, therefore, are not accessible then.

**DataBindingCollection**

*Example Syntax:*

**ToString**

[C#]	public		<b>DataBindingCollection();</b>
[C++]	public:		<b>DataBindingCollection();</b>
[VB]	Public	Sub	<b>New()</b>
[JScript]	public	function	<b>DataBindingCollection();</b>

*Description*

Initializes a new instance of the **System.Web.UI.DataBindingCollection** class.

**Count**

**ToString**

[C#]	public	int	<b>Count</b>	{get;}
[C++]	public:	__property	int	<b>get_Count();</b>
[VB]	Public	ReadOnly	Property	<b>Count As Integer</b>
[JScript]	public	function	get	<b>Count() : int;</b>

*Description*

Gets the number of **System.Web.UI.DataBinding** objects in the **System.Web.UI.DataBindingCollection** object.

IsReadOnly

ToString

[C#]            public            bool            IsReadOnly            {get;}

[C++]           public:            \_\_property           bool           get\_IsReadOnly();

[VB]   Public   ReadOnly   Property   IsReadOnly   As   Boolean

[JScript]   public   function   get   IsReadOnly()   :   Boolean;

*Description*

Gets a value indicating whether the **System.Web.UI.DataBindingCollection** is read-only.

IsSynchronized

ToString

[C#]            public            bool            IsSynchronized            {get;}

[C++]           public:            \_\_property           bool           get\_IsSynchronized();

[VB]   Public   ReadOnly   Property   IsSynchronized   As   Boolean

[JScript]   public   function   get   IsSynchronized()   :   Boolean;

*Description*

1 Gets a value indicating whether the  
2 **System.Web.UI.DataBindingCollection** is synchronized (thread-safe).

3 Item

4 ToString

6 [C#] public DataBinding this[string propertyName] {get;}

7 [C++] public: \_\_property DataBinding\* get\_Item(String\* propertyName);

8 [VB] Public Default ReadOnly Property Item(ByVal propertyName As String) As  
9 DataBinding

10 [JScript] returnValue = DataBindingCollectionObject.Item(propertyName);

12 *Description*

13 Gets the **System.Web.UI.DataBinding** object with the specified property  
14 name. The name of the property to be found.

15 RemovedBindings

16 ToString

18 [C#] public string[] RemovedBindings {get;}

19 [C++] public: \_\_property String\* get\_RemovedBindings();

20 [VB] Public ReadOnly Property RemovedBindings As String ()

21 [JScript] public function get RemovedBindings() : String[];

23 *Description*

24 Gets an array of the names of the **System.Web.UI.DataBinding** objects  
25 removed from the collection.

```

1      SyncRoot
2      ToString
3
4      [C#]          public          object          SyncRoot          {get;}
5      [C++]          public:          __property          Object*          get_SyncRoot();
6      [VB]          Public          ReadOnly          Property          SyncRoot          As          Object
7      [JScript]          public          function          get          SyncRoot()          :          Object;
8

```

#### 9 *Description*

10 Gets an object that can be used to synchronize access to the  
11 **System.Web.UI.DataBindingCollection** .

#### 12 Add

```

13
14      [C#]          public          void          Add(DataBinding          binding);
15      [C++]          public:          void          Add(DataBinding*          binding);
16      [VB]          Public          Sub          Add(ByVal          binding          As          DataBinding)
17      [JScript]          public          function          Add(binding          :          DataBinding);
18

```

#### 19 *Description*

20 Adds the specified **System.Web.UI.DataBinding** object to the  
21 **System.Web.UI.DataBindingCollection** . The data binding object to add.

#### 22 Clear

```

23
24      [C#]          public          void          Clear();
25      [C++]          public:          void          Clear();

```

```

1 [VB] Public Sub Clear()
2 [JScript] public function Clear();

```

3  
4 *Description*

5 Removes all **System.Web.UI.DataBinding** objects from the  
6 **System.Web.UI.DataBindingCollection**.

7 CopyTo

```

8
9 [C#] public void CopyTo(Array array, int index);
10 [C++] public: __sealed void CopyTo(Array* array, int index);
11 [VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As
12 Integer)
13 [JScript] public function CopyTo(array : Array, index : int);

```

14  
15 *Description*

16 Copies the **DataBindingCollection** values to a one-dimensional  
17 **System.Array**, beginning at the **Array** object's specified index. The one-  
18 dimensional **System.Array** object that is the destination of the values copied from  
19 **DataBindingCollection**. The index in the array, specified by the *array* parameter,  
20 where copying begins.

21 GetEnumerator

```

22
23 [C#] public IEnumerator GetEnumerator();
24 [C++] public: __sealed IEnumerator* GetEnumerator();
25 [VB] NotOverridable Public Function GetEnumerator() As IEnumerator

```



1 [JScript] public function GetEnumerator() : IEnumerator;

3 *Description*

4 Returns an enumerator to iterate through the  
5 **System.Web.UI.DataBindingCollection** object.

6 *Return Value:* An **System.Collections.IEnumerator** that contains the collection's  
7 members.

8 Remove

10 [C#] public void Remove(DataBinding binding);

11 [C++] public: void Remove(DataBinding\* binding);

12 [VB] Public Sub Remove(ByVal binding As DataBinding)

13 [JScript] public function Remove(binding : DataBinding);

15 *Description*

16 Removes the specified **System.Web.UI.DataBinding** object from the  
17 **System.Web.UI.DataBindingCollection** . The **System.Web.UI.DataBinding**  
18 object to be removed from the **System.Web.UI.DataBindingCollection**.

19 Remove

21 [C#] public void Remove(string propertyName);

22 [C++] public: void Remove(String\* propertyName);

23 [VB] Public Sub Remove(ByVal propertyName As String)

24 [JScript] public function Remove(propertyName : String); Removes a  
25 **System.Web.UI.DataBinding** object from the

## System.Web.UI.DataBindingCollection

### *Description*

Removes the **System.Web.UI.DataBinding** object associated with the specified property name from the **System.Web.UI.DataBindingCollection**. The property name associated with the **System.Web.UI.DataBinding** object to be removed.

#### Remove

```
[C#] public void Remove(string propertyName, bool addToRemovedList);
```

```
[C++] public: void Remove(String* propertyName, bool addToRemovedList);
```

```
[VB] Public Sub Remove(ByVal propertyName As String, ByVal  
addToRemovedList As Boolean)
```

```
[JScript] public function Remove(propertyName : String, addToRemovedList :  
Boolean);
```

### *Description*

Removes the **System.Web.UI.DataBinding** object, associated with the specified property name, from the **System.Web.UI.DataBindingCollection** and instructs the collection not to add the removed **DataBinding** object to the **System.Web.UI.DataBindingCollection.RemovedBindings** list. The property associated with the **DataBinding** object to be removed. A **Boolean** that indicates whether to add the property name to the **RemovedBindings** list.

DataBindingHandlerAttribute class (System.Web.UI)

ToString

1  
2  
3 *Description*

4 Specifies a design-time class that performs databinding of controls within a  
5 designer.

6 ToString

7  
8 [C#] public static readonly DataBindingHandlerAttribute Default;

9 [C++] public: static DataBindingHandlerAttribute\* Default;

10 [VB] Public Shared ReadOnly Default As DataBindingHandlerAttribute

11 [JScript] public static var Default : DataBindingHandlerAttribute;

12  
13 *Description*

14 Defines the default constructor for the  
15 **System.Web.UI.DataBindingHandlerAttribute** class. This field is read-only.

16 DataBindingHandlerAttribute

17 *Example Syntax:*

18 ToString

19  
20 [C#] public DataBindingHandlerAttribute();

21 [C++] public: DataBindingHandlerAttribute();

22 [VB] Public Sub New()

23 [JScript] public function DataBindingHandlerAttribute(); Initializes a new  
24 instance of the **System.Web.UI.DataBindingHandlerAttribute** class.

1  
2 *Description*

3       Initializes       a       new       instance       of       the  
4 **System.Web.UI.DataBindingHandlerAttribute** class using no parameters. This  
5 is the default constructor.

6       DataBindingHandlerAttribute

7       *Example Syntax:*

8       ToString

9  
10 [C#]       public       DataBindingHandlerAttribute(string       typeName);

11 [C++]       public:       DataBindingHandlerAttribute(String\*       typeName);

12 [VB]       Public       Sub       New(ByVal       typeName       As       String)

13 [JScript] public function DataBindingHandlerAttribute(typeName : String);

14  
15 *Description*

16       Initializes       a       new       instance       of       the  
17 **System.Web.UI.DataBindingHandlerAttribute** class with the specified type  
18 name.

19       The syntax for this attribute is: [DataBindingHandlerAttribute  
20 typeof((dataBindingHandlerType))] The name of the data binding handler's type.

21       DataBindingHandlerAttribute

22       *Example Syntax:*

23       ToString

24  
25 [C#]       public       DataBindingHandlerAttribute(Type       type);

```

1 [C++]      public:      DataBindingHandlerAttribute(Type*      type);
2 [VB]      Public      Sub      New(ByVal      type      As      Type)
3 [JScript] public      function      DataBindingHandlerAttribute(type      :      Type);
4

```

#### *Description*

Initializes a new instance of the **System.Web.UI.DataBindingHandlerAttribute** class of the specified Type. The type name for this constructor is the fully-qualified name of the type, including the assembly name from which it was loaded. The Type assigned to the data binding handler.

HandlerTypeName

ToString

```

14 [C#]      public      string      HandlerTypeName      {get;}
15 [C++]      public:      __property      String*      get_HandlerTypeName();
16 [VB]      Public      ReadOnly      Property      HandlerTypeName      As      String
17 [JScript] public      function      get      HandlerTypeName()      :      String;
18

```

#### *Description*

Gets the type name of the data binding handler. If the type name is **null**, this property returns an empty string.

TypeId

DataBoundLiteralControl class (System.Web.UI)

ToString

[illegible]

## DataBoundLiteralControl

ToString

```
[JScript] public function DataBoundLiteralControl(staticLiteralsCount : int,
dataBoundLiteralCount : int);
```

ClientID

1	Context
2	Controls
3	EnableViewState
4	Events
5	HasChildViewState
6	ID
7	IsTrackingViewState
8	NamingContainer
9	Page
10	Parent
11	Site
12	TemplateSourceDirectory
13	Text
14	ToString
15	
16	
17	<i>Description</i>
18	Gets or sets the text content of the
19	<b>System.Web.UI.DataBoundLiteralControl</b> object.
20	UniqueID
21	ViewState
22	ViewStateIgnoresCase
23	Visible
24	CreateControlCollection
25	

```

1
2 [C#]    protected    override    ControlCollection    CreateControlCollection();
3 [C++]    protected:    ControlCollection*    CreateControlCollection();
4 [VB]    Overrides    Protected    Function    CreateControlCollection()    As
5 ControlCollection
6 [JScript]    protected    override    function    CreateControlCollection()    :
7 ControlCollection;
8

```

#### 9 *Description*

10 Creates an **System.Web.UI.EmptyCollectionControl** object for the  
11 current instance of the **System.Web.UI.DataBoundLiteralControl**.

12 *Return Value:* The **EmptyCollectionControl** object for the current control.

#### 13 **LoadViewState**

```

14
15 [C#]    protected    override    void    LoadViewState(object    savedState);
16 [C++]    protected:    void    LoadViewState(Object*    savedState);
17 [VB]    Overrides    Protected    Sub    LoadViewState(ByVal    savedState    As    Object)
18 [JScript]    protected    override    function    LoadViewState(savedState    :    Object);
19

```

#### 20 *Description*

21 Loads the previously saved view state. You can override this method to  
22 synchronize **System.Web.UI.DataBoundLiteralControl.Text** property with new  
23 literal content. The previously saved view state.

#### 24 **Render**



```

1
2 [C#]    protected    override    void    Render(HtmlTextWriter    output);
3 [C++]    protected:    void    Render(HtmlTextWriter*    output);
4 [VB]    Overrides    Protected    Sub    Render(ByVal    output    As    HtmlTextWriter)
5 [JScript]    protected    override    function    Render(output    :    HtmlTextWriter);
6

```

#### 7 *Description*

8 Writes the content of the **System.Web.UI.DataBoundLiteralControl**  
9 object to an output stream. The output stream that renders HTML content to the  
10 client.

#### 11 **SaveViewState**

```

12
13 [C#]    protected    override    object    SaveViewState();
14 [C++]    protected:    Object*    SaveViewState();
15 [VB]    Overrides    Protected    Function    SaveViewState()    As    Object
16 [JScript]    protected    override    function    SaveViewState()    :    Object;
17

```

#### 18 *Description*

19 Saves any view state modified after the control began monitoring state  
20 changes.

21 *Return Value:* An **System.Object** that contains the saved view state values.

#### 22 **SetDataBoundString**

```

23
24 [C#]    public    void    SetDataBoundString(int    index,    string    s);
25 [C++]    public:    void    SetDataBoundString(int    index,    String*    s);

```

1 [VB] Public Sub SetDataBoundString(ByVal index As Integer, ByVal s As String)

2 [JScript] public function SetDataBoundString(index : int, s : String);

4 *Description*

6 SetStaticString

8 [C#] public void SetStaticString(int index, string s);

9 [C++] public: void SetStaticString(int index, String\* s);

10 [VB] Public Sub SetStaticString(ByVal index As Integer, ByVal s As String)

11 [JScript] public function SetStaticString(index : int, s : String);

13 *Description*

15 DesignerDataBoundLiteralControl class (System.Web.UI)

16 TrackViewState

19 *Description*

20 Simpler version of DataBoundLiteralControlBuilder, used at design time.

21 DesignerDataBoundLiteralControl

22 *Example Syntax:*

23 TrackViewState

25 [C#] public DesignerDataBoundLiteralControl();

1	[C++]	public:	DesignerDataBoundLiteralControl();
2	[VB]	Public	Sub New()
3	[JScript]	public function	DesignerDataBoundLiteralControl();

4

5 *Description*

6

7 ChildControlsCreated

8 ClientID

9 Context

10 Controls

11 EnableViewState

12 Events

13 HasChildViewState

14 ID

15 IsTrackingViewState

16 NamingContainer

17 Page

18 Parent

19 Site

20 TemplateSourceDirectory

21 Text

22 TrackViewState

23

24

25 *Description*

TOP OF PAGE

1 Gets or sets the text content of the data-bound literal control.  
 2 UniqueID  
 3 ViewState  
 4 ViewStateIgnoresCase  
 5 Visible  
 6 CreateControlCollection

8 [C#] protected override ControlCollection CreateControlCollection();  
 9 [C++] protected: ControlCollection\* CreateControlCollection();  
 10 [VB] Overrides Protected Function CreateControlCollection() As  
 11 ControlCollection  
 12 [JScript] protected override function CreateControlCollection() :  
 13 ControlCollection;

14  
 15 *Description*

16  
 17 LoadViewState

18  
 19 [C#] protected override void LoadViewState(object savedState);  
 20 [C++] protected: void LoadViewState(Object\* savedState);  
 21 [VB] Overrides Protected Sub LoadViewState(ByVal savedState As Object)  
 22 [JScript] protected override function LoadViewState(savedState : Object);

23  
 24 *Description*  
 25

1 Loads the previously saved state. Overridden to synchronize Text property  
2 with LiteralContent.

### 3 Render

4  
5 [C#] protected override void Render(HtmlTextWriter output);

6 [C++] protected: void Render(HtmlTextWriter\* output);

7 [VB] Overrides Protected Sub Render(ByVal output As HtmlTextWriter)

8 [JScript] protected override function Render(output : HtmlTextWriter);

### 9 10 *Description*

11 Saves any state that was modified after the control began monitoring state  
12 changes.

### 13 SaveViewState

14  
15 [C#] protected override object SaveViewState();

16 [C++] protected: Object\* SaveViewState();

17 [VB] Overrides Protected Function SaveViewState() As Object

18 [JScript] protected override function SaveViewState() : Object;

### 19 20 *Description*

21 The object that contains the state changes.

22 DesignTimeParseData class (System.Web.UI)

23 TrackViewState

24 DesignTimeParseData

25 *Example Syntax:*

```

1      TrackViewState
2      DataBindingHandler
3      TrackViewState
4      DesignerHost
5      TrackViewState
6      DocumentUrl
7      TrackViewState
8      ParseText
9      TrackViewState
10     DesignTimeTemplateParser class (System.Web.UI)
11     ToString
12     ParseControl
13
14     [C#]   public   static   Control   ParseControl(DesignTimeParseData   data);
15     [C++]  public:  static   Control*   ParseControl(DesignTimeParseData*   data);
16     [VB]   Public Shared Function ParseControl(ByVal data As DesignTimeParseData)
17     As                                           Control
18     [JScript] public static function ParseControl(data : DesignTimeParseData) :
19     Control;
20
21     ParseTemplate
22
23     [C#]   public   static   ITemplate   ParseTemplate(DesignTimeParseData   data);
24     [C++]  public:  static   ITemplate*   ParseTemplate(DesignTimeParseData*   data);
25     [VB]   Public   Shared   Function   ParseTemplate(ByVal   data   As
DesignTimeParseData)           As           ITemplate

```

1 [JScript] public static function ParseTemplate(data : DesignTimeParseData) :  
2 ITemplate;

3 EmptyControlCollection class (System.Web.UI)

4 ToString

7 *Description*

8 Provides standard support for a ControlCollection that is always empty.

9 This class is used when you want to define a custom control that does not  
10 allow child controls.

11 EmptyControlCollection

12 *Example Syntax:*

13 ToString

15 [C#] public EmptyControlCollection(Control owner);

16 [C++] public: EmptyControlCollection(Control\* owner);

17 [VB] Public Sub New(ByVal owner As Control)

18 [JScript] public function EmptyControlCollection(owner : Control);

20 *Description*

21 Instantiates an empty control collection.

22 Count

23 IsReadOnly

24 IsSynchronized

25 Item

Owner

SyncRoot

Add

[C#] public override void Add(Control child);

[C++] public: void Add(Control\* child);

[VB] Overrides Public Sub Add(ByVal child As Control)

[JScript] public override function Add(child : Control);

*Description*

Denies the addition of the specified **System.Web.UI.Control** object to the collection.

AddAt

[C#] public override void AddAt(int index, Control child);

[C++] public: void AddAt(int index, Control\* child);

[VB] Overrides Public Sub AddAt(ByVal index As Integer, ByVal child As Control)

[JScript] public override function AddAt(index : int, child : Control);

*Description*

Denies the addition of the specified **System.Web.UI.Control** object to the collection, at the specified index position.

Html32TextWriter class (System.Web.UI)

ToString



1  
2  
3 *Description*

4 Provides a text writer for ASP.NET server controls that render content to  
5 downlevel clients only.

6 `Html32TextWriter`

7 *Example Syntax:*

8 `ToString`

9 `System.Web.UI.Html32TextWriter`

10  
11 *Description*

12 Initializes a new instance of the `System.Web.UI.Html32TextWriter` class  
13 that uses the `System.Web.UI.HtmlTextWriter.DefaultTabString` constant when  
14 indentation of a line is necessary. The `System.IO.TextWriter` object to render the  
15 HTML content.

16 `Html32TextWriter`

17 *Example Syntax:*

18 `ToString`

19  
20 [C#] public Html32TextWriter(TextWriter writer, string tabString);

21 [C++] public: Html32TextWriter(TextWriter\* writer, String\* tabString);

22 [VB] Public Sub New(ByVal writer As TextWriter, ByVal tabString As String)

23 [JScript] public function Html32TextWriter(writer : TextWriter, tabString :  
24 String);

### Description

Initializes a new instance of the **System.Web.UI.Html32TextWriter** class using the specified tab spacing. The **System.IO.TextWriter** object to render the HTML content. A **System.String** that represents the number of spaces defined in the **System.Web.UI.HtmlTextWriter.Indent** property.

## Encoding

FontStack

ToString

### Description

Gets a collection of font information for the HTML to render.

## FormatProvider

## Indent

# InnerWriter

NewLine

## TagKey

## TagName

## GetTagName

```
[C#] protected override string GetTagName(HtmlTextWriterTag tagKey);
```

```
[C++]    protected:    String*    GetTagName(HtmlTextWriterTag    tagKey);
```

[VB]	Overrides	Protected	Function	GetTagName(ByVal tagKey As	
	HtmlTextWriterTag)			As	String

1 [JScript] protected override function GetTagName(tagKey : HtmlTextWriterTag) :  
2 String;

3  
4 *Description*

5 Obtains the HTML element associated with the specified  
6 **System.Web.UI.HtmlTextWriterTag** enumeration value.

7 *Return Value:* The HTML element.

8 If HtmlTextWriterTag.Div is passed in the tagKey parameter, this method  
9 returns the HTML  
10 element. The **HtmlTextWriterTag** value to obtain the HTML element for.

11 **OnStyleAttributeRender**

12  
13 [C#] protected override bool OnStyleAttributeRender(string name, string value,  
14 HtmlTextWriterStyle key);

15 [C++] protected: bool OnStyleAttributeRender(String\* name, String\* value,  
16 HtmlTextWriterStyle key);

17 [VB] Overrides Protected Function OnStyleAttributeRender(ByVal name As  
18 String, ByVal value As String, ByVal key As HtmlTextWriterStyle) As Boolean

19 [JScript] protected override function OnStyleAttributeRender(name : String, value  
20 : String, key : HtmlTextWriterStyle) : Boolean;

21  
22 *Description*

23 Determines whether the specified HTML style attribute and its value have  
24 been rendered to the requesting page.

1 *Return Value:* **true** if the HTML style attribute and its value have been rendered to  
2 the requesting page; otherwise, **false** .

3 This method supports a smaller number of HTML style attributes than the  
4 of the  
5 **System.Web.UI.HtmlTextWriter.OnStyleAttributeRender(System.String, System.String, System.Web.UI.HtmlTextWriterStyle)** method which it overrides.  
6 The HTML style attribute to render to the client. The value associated with the  
7 HTML style attribute. The **System.Web.UI.HtmlTextWriterStyle** enumeration  
8 value associated with the HTML style attribute.

#### 10 OnTagRender

12 [C#] protected override bool OnTagRender(string name, HtmlTextWriterTag key);

13 [C++] protected: bool OnTagRender(String\* name, HtmlTextWriterTag key);

14 [VB] Overrides Protected Function OnTagRender(ByVal name As String, ByVal

15 key As HtmlTextWriterTag) As Boolean

16 [JScript] protected override function OnTagRender(name : String, key :

17 HtmlTextWriterTag) : Boolean;

#### 19 Description

20 Determines whether the specified HTML element has been rendered to the  
21 requesting page.

22 *Return Value:* **true** if the HTML element has been rendered to the requesting  
23 page; otherwise **false** .

24 This method associates the

1 element with an HTML table to simplify page layout for browsers that do  
2 not support this element. The HTML element to render. The  
3 **System.Web.UI.HtmlTextWriterTag** enumeration value associated with the  
4 HTML element.

#### 5 RenderAfterContent

6  
7 [C#] protected override string RenderAfterContent();  
8 [C++] protected: String\* RenderAfterContent();  
9 [VB] Overrides Protected Function RenderAfterContent() As String  
10 [JScript] protected override function RenderAfterContent() : String;  
11

#### 12 *Description*

13 Writes any text or spacing that occurs after the content of the HTML  
14 element to the **System.Web.UI.HtmlTextWriter** output stream.

15 *Return Value:* The spacing or text to write prior to the content of the HTML  
16 element. If there is no such information to render, this method returns **null** .

#### 17 RenderAfterTag

18  
19 [C#] protected override string RenderAfterTag();  
20 [C++] protected: String\* RenderAfterTag();  
21 [VB] Overrides Protected Function RenderAfterTag() As String  
22 [JScript] protected override function RenderAfterTag() : String;  
23

#### 24 *Description*

25

1 Writes any spacing or text that occurs after an HTML element's closing tag.  
2 *Return Value:* The spacing or text to write after the closing tag of the HTML  
3 element. If there is no such information to render, this method returns **null**.

#### 4 RenderBeforeContent

5  
6 [C#] protected override string RenderBeforeContent();  
7 [C++] protected: String\* RenderBeforeContent();  
8 [VB] Overrides Protected Function RenderBeforeContent() As String  
9 [JScript] protected override function RenderBeforeContent() : String;

#### 11 Description

12 Writes any tab spacing or font information that appears before the content  
13 contained in an HTML element.  
14 *Return Value:* The font information or spacing to write prior to the content of the  
15 HTML element. If there is no such information to render, this method returns **null**.

#### 17 RenderBeforeTag

18  
19 [C#] protected override string RenderBeforeTag();  
20 [C++] protected: String\* RenderBeforeTag();  
21 [VB] Overrides Protected Function RenderBeforeTag() As String  
22 [JScript] protected override function RenderBeforeTag() : String;

#### 24 Description

Writes any text or tab spacing that occurs before the opening tag of an HTML element to the **System.Web.UI.HtmlTextWriter** output stream.

*Return Value:* Any HTML font and spacing information to render before the tag; if there is no such information to render, this method returns **null**.

#### RenderBeginTag

```
[C#] public override void RenderBeginTag(HtmlTextWriterTag tagKey);
[C++] public: void RenderBeginTag(HtmlTextWriterTag tagKey);
[VB] Overrides Public Sub RenderBeginTag(ByVal tagKey As
HtmlTextWriterTag)
[JScript] public override function RenderBeginTag(tagKey : HtmlTextWriterTag);
```

#### Description

Writes the opening tag of the specified HTML element to the **System.Web.UI.HtmlTextWriter** output stream.

If the element is specified, this method performs basic HTML formatting to present the HTML content. The **System.Web.UI.HtmlTextWriterTag** enumeration value that indicates the HTML element to write.

#### RenderEndTag

```
[C#] public override void RenderEndTag();
[C++] public: void RenderEndTag();
[VB] Overrides Public Sub RenderEndTag()
```

1 [JScript] public override function RenderEndTag();

2  
3 *Description*

4 Writes the end tag of an HTML element to the  
5 **System.Web.UI.HtmlTextWriter** output stream along with any font information  
6 that is associated with the element.

7 HtmlTextWriter class (System.Web.UI)

8 WriteStyleAttribute

9  
10  
11 *Description*

12 Represents a writer that writes a sequential series of HTML-specific  
13 characters and text. This class provides formatting capabilities that ASP.NET  
14 server controls use when rendering HTML content to clients.

15 This class is commonly used to render HTML to a Web Forms page. Use it  
16 as an output stream when you override the  
17 **System.Web.UI.Control.Render(System.Web.UI.HtmlTextWriter)** ,  
18 **System.Web.UI.LiteralControl.Render(System.Web.UI.HtmlTextWriter)** ,  
19 and other ASP.NET server control methods that write content to a Web page.

20 WriteStyleAttribute

21  
22  
23 *Description*

24 Represents the characters that make up the default tab.

25 WriteStyleAttribute



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#]      public      const      char      DoubleQuoteChar;  
[C++]     public:     const      __wchar_t  DoubleQuoteChar;  
[VB]      Public      Const      DoubleQuoteChar      As      Char  
[JScript] public      var      DoubleQuoteChar      :      Char;
```

*Description*

Represents a double-quote character.

WriteStyleAttribute

```
[C#]      public      const      string      EndTagLeftChars;  
[C++]     public:     const      String*      EndTagLeftChars;  
[VB]      Public      Const      EndTagLeftChars      As      String  
[JScript] public      var      EndTagLeftChars      :      String;
```

*Description*

Represents the left angle bracket and closing forward slash for a closing tag of an HTML element.

WriteStyleAttribute

```
[C#]      public      const      char      EqualsChar;  
[C++]     public:     const      __wchar_t  EqualsChar;  
[VB]      Public      Const      EqualsChar      As      Char  
[JScript] public      var      EqualsChar      :      Char;
```

*Description*

Represents the equal sign character.

WriteStyleAttribute

[C#]        public        const        string        EqualsDoubleQuoteString;

[C++]       public:        const        String\*       EqualsDoubleQuoteString;

[VB]        Public        Const        EqualsDoubleQuoteString       As       String

[JScript]   public        var        EqualsDoubleQuoteString       :       String;

*Description*

Represents an equal sign, a forward slash, and a double quote character together in a **System.String** .

WriteStyleAttribute

[C#]        public        const        string        SelfClosingChars;

[C++]       public:        const        String\*       SelfClosingChars;

[VB]        Public        Const        SelfClosingChars       As       String

[JScript]   public        var        SelfClosingChars       :       String;

*Description*

Represents the self-closing forward slash character of an HTML tag.

This charcter is used in HTML elements that are self-closed. For example, .

WriteStyleAttribute

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

[C#]	public	const	string	SelfClosingTagEnd;
[C++]	public:	const	String*	SelfClosingTagEnd;
[VB]	Public	Const	SelfClosingTagEnd	As String
[JScript]	public	var	SelfClosingTagEnd	: String;

*Description*

Represents the closing forward slash and right angle bracket of a self-closing HTML element.

WriteStyleAttribute

[C#]	public	const	char	SemicolonChar;
[C++]	public:	const	__wchar_t	SemicolonChar;
[VB]	Public	Const	SemicolonChar	As Char
[JScript]	public	var	SemicolonChar	: Char;

*Description*

Represents the semicolon character.

WriteStyleAttribute

[C#]	public	const	char	SingleQuoteChar;
[C++]	public:	const	__wchar_t	SingleQuoteChar;
[VB]	Public	Const	SingleQuoteChar	As Char
[JScript]	public	var	SingleQuoteChar	: Char;

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

*Description*

Represents a single quote character.

WriteStyleAttribute

[C#]	public	const	char	SlashChar;
[C++]	public:	const	__wchar_t	SlashChar;
[VB]	Public	Const	SlashChar	As Char
[JScript]	public	var	SlashChar	: Char;

*Description*

Represents the backslash character.

WriteStyleAttribute

[C#]	public	const	char	SpaceChar;
[C++]	public:	const	__wchar_t	SpaceChar;
[VB]	Public	Const	SpaceChar	As Char
[JScript]	public	var	SpaceChar	: Char;

*Description*

Represents a space character.

WriteStyleAttribute

[C#]	public	const	char	StyleEqualsChar;
[C++]	public:	const	__wchar_t	StyleEqualsChar;

1	[VB]	Public	Const	StyleEqualsChar	As	Char
2	[JScript]	public	var	StyleEqualsChar	:	Char;

3

4 *Description*

5       Represents the colon character.

6       WriteStyleAttribute

8	[C#]	public	const	char		TagLeftChar;
9	[C++]	public:	const	__wchar_t		TagLeftChar;
10	[VB]	Public	Const	TagLeftChar	As	Char
11	[JScript]	public	var	TagLeftChar	:	Char;

12

13 *Description*

14       Represents the opening angle-bracket character of an HTML tag.

15       WriteStyleAttribute

17	[C#]	public	const	char		TagRightChar;
18	[C++]	public:	const	__wchar_t		TagRightChar;
19	[VB]	Public	Const	TagRightChar	As	Char
20	[JScript]	public	var	TagRightChar	:	Char;

21

22 *Description*

23       Represents the closing angle-bracket character of an HTML tag.

24       HtmlTextWriter

25       *Example Syntax:*

## WriteStyleAttribute

[C#]            public            HtmlTextWriter(TextWriter            writer);

[C++]           public:            HtmlTextWriter(TextWriter\*            writer);

[VB]        Public        Sub        New(ByVal        writer        As        TextWriter)

[JScript] public function HtmlTextWriter(writer : TextWriter); Initializes a new instance of the **System.Web.UI.HtmlTextWriter** class.

### *Description*

Initializes a new instance of the **System.Web.UI.HtmlTextWriter** class that uses the **System.Web.UI.HtmlTextWriter.DefaultTabString** constant when indentation of a line is necessary. The **System.IO.TextWriter** object to render the HTML content.

HtmlTextWriter

### *Example Syntax:*

WriteStyleAttribute

[C#]    public    HtmlTextWriter(TextWriter    writer,    string    tabString);

[C++]    public:    HtmlTextWriter(TextWriter\*    writer,    String\*    tabString);

[VB] Public Sub New(ByVal writer As TextWriter, ByVal tabString As String)

[JScript] public function HtmlTextWriter(writer : TextWriter, tabString : String);

### *Description*

Initializes a new instance of the **System.Web.UI.HtmlTextWriter** class with the line indention as specified in the *tabString* parameter. The

**System.IO.TextWriter** object to render the HTML content. A **System.String** that represents the number of spaces defined in the **System.Web.UI.HtmlTextWriter.Indent** property.

Encoding

WriteStyleAttribute

[C#] public override Encoding Encoding {get;}

[C++] public: \_\_property virtual Encoding\* get\_Encoding();

[VB] Overrides Public ReadOnly Property Encoding As Encoding

[JScript] public function get Encoding() : Encoding;

#### *Description*

Gets the **System.Text.Encoding** that the **System.Web.UI.HtmlTextWriter** object uses to write content to the page.

FormatProvider

Indent

WriteStyleAttribute

#### *Description*

Gets or sets the number of spaces to indent at the beginning of a line.

InnerWriter

WriteStyleAttribute

[C#] public TextWriter InnerWriter {get; set;}

1 [C++] public: \_\_property TextWriter\* get\_InnerWriter();public: \_\_property void  
2 set\_InnerWriter(TextWriter\*);

3 [VB] Public Property InnerWriter As TextWriter

4 [JScript] public function get InnerWriter() : TextWriter;public function set  
5 InnerWriter(TextWriter);

6  
7 *Description*

8 Gets the text writer that writes the inner content of the HTML element.

9 NewLine

10 WriteStyleAttribute

11  
12 [C#] public override string NewLine {get; set;}

13 [C++] public: \_\_property virtual String\* get\_NewLine();public: \_\_property virtual  
14 void set\_NewLine(String\*);

15 [VB] Overrides Public Property NewLine As String

16 [JScript] public function get NewLine() : String;public function set  
17 NewLine(String);

18  
19 *Description*

20 Gets or sets the line terminator string used by the current  
21 **System.Web.UI.HtmlTextWriter** .

22 The default line terminator string is a carriage return followed by a line feed  
23 ("r\n").

24 TagKey

25 WriteStyleAttribute



```

1
2 [C#]      protected      HtmlTextWriterTag      TagKey      {get;      set;}
3 [C++]  protected:  __property  HtmlTextWriterTag  get_TagKey();protected:
4  __property          void          set_TagKey(HtmlTextWriterTag);
5 [VB]      Protected      Property      TagKey      As      HtmlTextWriterTag
6 [JScript] protected function get TagKey() : HtmlTextWriterTag;protected function
7 set          TagKey(HtmlTextWriterTag);
8

```

### *Description*

Gets or sets the **System.Web.UI.HtmlTextWriterTag** value for the specified HTML element.

TagName

WriteStyleAttribute

```

14
15 [C#]      protected      string      TagName      {get;      set;}
16 [C++]  protected:  __property  String*  get_TagName();protected:  __property void
17 set_TagName(String*);
18 [VB]      Protected      Property      TagName      As      String
19 [JScript] protected function get TagName() : String;protected function set
20 TagName(String);
21

```

### *Description*

Gets or sets the name of the HTML element to be written to the rendered page.

AddAttribute

```

1 [C#] public virtual void AddAttribute(HtmlTextWriterAttribute key, string value);
2
3 [C++] public: virtual void AddAttribute(HtmlTextWriterAttribute key, String*
4 value);
5
6 [VB] Overridable Public Sub AddAttribute(ByVal key As
7 HtmlTextWriterAttribute, ByVal value As String)
8
9 [JScript] public function AddAttribute(key : HtmlTextWriterAttribute, value :
10 String);
11

```

### *Description*

Adds the HTML attribute associated with the specified **System.Web.UI.HtmlTextWriterAttribute** key and the specified value to the **System.Web.UI.HtmlTextWriter** output stream. An **HtmlTextWriterAttribute** value that represents the HTML attribute. The value to assign to the HTML attribute.

### **AddAttribute**

```

17
18 [C#] public virtual void AddAttribute(string name, string value);
19
20 [C++] public: virtual void AddAttribute(String* name, String* value);
21
22 [VB] Overridable Public Sub AddAttribute(ByVal name As String, ByVal value
23 As String)
24
25 [JScript] public function AddAttribute(name : String, value : String); Adds an
HTML attribute and its value to an System.Web.UI.HtmlTextWriter output
stream to be rendered on a client.

```

1  
2 *Description*

3 Adds the specified HTML attribute and value to the **HtmlTextWriter**  
4 output stream. The HTML attribute to add. The value to assign to the HTML  
5 attribute.

6 **AddAttribute**

7  
8 [C#] public virtual void AddAttribute(HtmlTextWriterAttribute key, string value,  
9 bool fEncode);

10 [C++] public: virtual void AddAttribute(HtmlTextWriterAttribute key, String\*  
11 value, bool fEncode);

12 [VB] Overridable Public Sub AddAttribute(ByVal key As  
13 HtmlTextWriterAttribute, ByVal value As String, ByVal fEncode As Boolean)

14 [JScript] public function AddAttribute(key : HtmlTextWriterAttribute, value :  
15 String, fEncode : Boolean);

16  
17 *Description*

18 Adds the HTML attribute associated with the  
19 **System.Web.UI.HtmlTextWriterAttribute** value specified by the key parameter,  
20 the specified attribute value, and a value that indicates if the attribute and value  
21 should be HTML encoded to the **System.Web.UI.HtmlTextWriter** output  
22 stream. An **HtmlTextWriterAttribute** value that represents the HTML attribute.  
23 The value to assign to the HTML attribute. **true** to HTML encode the attribute and  
24 its value; otherwise, **false**.

25 **AddAttribute**

1  
2 [C#] public virtual void AddAttribute(string name, string value, bool fEncode);  
3 [C++] public: virtual void AddAttribute(String\* name, String\* value, bool  
4 fEncode);  
5 [VB] Overridable Public Sub AddAttribute(ByVal name As String, ByVal value  
6 As String, ByVal fEncode As Boolean)  
7 [JScript] public function AddAttribute(name : String, value : String, fEncode :  
8 Boolean);  
9

### 10 *Description*

11 Adds the specified HTML attribute and value to the **HtmlTextWriter**  
12 output stream, with a value that indicates if the attribute and value should be  
13 HTML encoded. The HTML attribute to add. The value to assign to the HTML  
14 attribute. **true** to HTML encode the attribute and its value; otherwise, **false**.

### 15 *AddAttribute*

16  
17 [C#] protected virtual void AddAttribute(string name, string value,  
18 HtmlTextWriterAttribute key);  
19 [C++] protected: virtual void AddAttribute(String\* name, String\* value,  
20 HtmlTextWriterAttribute key);  
21 [VB] Overridable Protected Sub AddAttribute(ByVal name As String, ByVal  
22 value As String, ByVal key As HtmlTextWriterAttribute)  
23 [JScript] protected function AddAttribute(name : String, value : String, key :  
24 HtmlTextWriterAttribute);  
25

## Description

Adds the specified HTML attribute and its value, along with an **System.Web.UI.HtmlTextWriterAttribute** value, to the **HtmlTextWriter** output stream. The HTML attribute to add. The value to assign to the HTML attribute. An **HtmlTextWriterAttribute** value that represents the HTML attribute.

### AddStyleAttribute

```
[C#] public virtual void AddStyleAttribute(HtmlTextWriterStyle key, string value);
```

```
[C++] public: virtual void AddStyleAttribute(HtmlTextWriterStyle key, String* value);
```

```
[VB] Overridable Public Sub AddStyleAttribute(ByVal key As HtmlTextWriterStyle, ByVal value As String)
```

```
[JScript] public function AddStyleAttribute(key : HtmlTextWriterStyle, value : String);
```

## Description

Adds the HTML style attribute associated with the **System.Web.UI.HtmlTextWriterStyle** value specified by the *key* parameter and the attribute's value to the **System.Web.UI.HtmlTextWriter** output stream. An **HtmlTextWriterStyle** value that represents the HTML style attribute to add. The value to assign to the HTML attribute.

### AddStyleAttribute

```

1
2 [C#] public virtual void AddStyleAttribute(string name, string value);
3 [C++] public: virtual void AddStyleAttribute(String* name, String* value);
4 [VB] Overridable Public Sub AddStyleAttribute(ByVal name As String, ByVal
5 value                                     As                                     String)
6 [JScript] public function AddStyleAttribute(name : String, value : String); Adds an
7 HTML style attribute to the to the System.Web.UI.HtmlTextWriter output
8 stream.
9

```

#### *Description*

Adds the specified HTML style attribute and its value to the **System.Web.UI.HtmlTextWriter** output stream. The HTML style attribute to add. The value assigned to the HTML style attribute.

#### **AddStyleAttribute**

```

16 [C#] protected virtual void AddStyleAttribute(string name, string value,
17 HtmlTextWriterStyle                                     key);
18 [C++] protected: virtual void AddStyleAttribute(String* name, String* value,
19 HtmlTextWriterStyle                                     key);
20 [VB] Overridable Protected Sub AddStyleAttribute(ByVal name As String, ByVal
21 value As String, ByVal key As HtmlTextWriterStyle)
22 [JScript] protected function AddStyleAttribute(name : String, value : String, key :
23 HtmlTextWriterStyle);
24

```

#### *Description*

Adds the specified HTML style attribute, along with its value, to the **System.Web.UI.HtmlTextWriter** output stream. The HTML style attribute to add to the output stream. The value to assign to the HTML attribute. An **System.Web.UI.HtmlTextWriterStyle** value that represents the HTML style attribute to add.

Close

[C#]	public	override	void	Close();
[C++]	public:		void	Close();
[VB]	Overrides	Public	Sub	Close()
[JScript]	public	override	function	Close();

### *Description*

Closes the current **System.Web.UI.HtmlTextWriter** and releases any system resources associated with it.

This implementation of **Close** calls the **System.IO.TextWriter** method passing a **true** value.

EncodeAttributeValue

[C#]	protected virtual string	EncodeAttributeValue(HtmlTextWriterAttribute
	attrKey,	string value);
[C++]	protected: virtual String*	EncodeAttributeValue(HtmlTextWriterAttribute
	attrKey,	String* value);
[VB]	Overridable Protected Function	EncodeAttributeValue(ByVal attrKey As
	HtmlTextWriterAttribute,	ByVal value As String) As String

[JScript]      protected      function      EncodeAttributeValue(attrKey      :  
 HtmlTextWriterAttribute,      value      :      String)      :      String;

*Description*

HTML encodes the specified HTML attribute's value.

*Return Value:* The encoded attribute value. An **System.Web.UI.HtmlTextWriterAttribute** value representing the HTML attribute to which the *value* parameter is assigned. The value assigned to the specified HTML attribute.

EncodeAttributeValue

[C#]    protected    string    EncodeAttributeValue(string    value,    bool    fEncode);

[C++]    protected:    String\*    EncodeAttributeValue(String\*    value,    bool    fEncode);

[VB]    Protected    Function    EncodeAttributeValue(ByVal    value    As    String,    ByVal  
 fEncode      As      Boolean)      As      String

[JScript]    protected    function    EncodeAttributeValue(value : String, fEncode :  
 Boolean) : String; HTML encodes the specified HTML attribute's value.

*Description*

HTML encodes the specified HTML attribute's value.

*Return Value:* The HTML-encoded attribute value, **null** if the *value* parameter is empty, or the unencoded attribute value if *fEncode* is **false** . The attribute value to encode. **true** to HTML encode the attribute value; otherwise, **false**.

EncodeUrl



```

1
2 [C#]          protected          string          EncodeUrl(string          url);
3 [C++]          protected:          String*          EncodeUrl(String*          url);
4 [VB] Protected Function EncodeUrl(ByVal url As String) As String
5 [JScript] protected function EncodeUrl(url : String) : String;
6

```

### Description

Performs minimal URL encoding by converting spaces passed in the *url* parameter to "%20".

*Return Value:* The encoded URL. The URL to be encoded.

### FilterAttributes

```

11
12
13 [C#]          protected          virtual          void          FilterAttributes();
14 [C++]          protected:          virtual          void          FilterAttributes();
15 [VB]          Overridable          Protected          Sub          FilterAttributes()
16 [JScript]          protected          function          FilterAttributes();
17

```

### Description

Renders all HTML attributes and style attributes by calling **System.Web.UI.HtmlTextWriter.OnAttributeRender(System.String,System.String,System.Web.UI.HtmlTextWriterAttribute)** and **System.Web.UI.HtmlTextWriter.OnStyleAttributeRender(System.String,System.String,System.Web.UI.HtmlTextWriterStyle)** on all properties of the page or ASP.NET server control.

### Flush

1					
2	[C#]	public	override	void	Flush();
3	[C++]	public:		void	Flush();
4	[VB]	Overrides	Public	Sub	Flush()
5	[JScript]	public	override	function	Flush();

6

7 *Description*

8        Clears all buffers for the current **System.Web.UI.HtmlTextWriter** and

9 causes any buffered data to be written to the text stream.

10        GetAttributeKey

11

12 [C#]    protected    HtmlTextWriterAttribute    GetAttributeKey(string    attrName);

13 [C++]    protected:    HtmlTextWriterAttribute    GetAttributeKey(String\*    attrName);

14 [VB]    Protected    Function    GetAttributeKey(ByVal    attrName    As    String)    As

15        HtmlTextWriterAttribute

16 [JScript]    protected    function    GetAttributeKey(attrName    :    String)    :

17        HtmlTextWriterAttribute;

18

19 *Description*

20        Obtains the corresponding **System.Web.UI.HtmlTextWriterAttribute**

21 enumeration value for the specified HTML attribute.

22 *Return Value:* The **HtmlTextWriterAttribute** enumeration value for the specified

23 HTML attribute. The HTML attribute to obtain the **HtmlTextWriterAttribute**

24 value for.

25        GetAttributeName

```

1
2 [C#] protected string GetAttributeName(HtmlTextWriterAttribute attrKey);
3 [C++] protected: String* GetAttributeName(HtmlTextWriterAttribute attrKey);
4 [VB] Protected Function GetAttributeName(ByVal attrKey As
5 HtmlTextWriterAttribute) As String
6 [JScript] protected function GetAttributeName(attrKey :
7 HtmlTextWriterAttribute) : String;

```

#### 9 *Description*

10 Obtains the name of the HTML attribute associated with the specified  
11 **System.Web.UI.HtmlTextWriterAttribute** value.

12 *Return Value:* The name of the HTML attribute. The **HtmlTextWriterAttribute**  
13 to obtain the HTML attribute name for.

#### 14 *GetStyleKey*

```

15
16 [C#] protected HtmlTextWriterStyle GetStyleKey(string styleName);
17 [C++] protected: HtmlTextWriterStyle GetStyleKey(String* styleName);
18 [VB] Protected Function GetStyleKey(ByVal styleName As String) As
19 HtmlTextWriterStyle
20 [JScript] protected function GetStyleKey(styleName : String) :
21 HtmlTextWriterStyle;

```

#### 23 *Description*

24 Obtains the **System.Web.UI.HtmlTextWriterStyle** enumeration value for  
25 the specified HTML style.

1 *Return Value:* The **HtmlTextWriterStyle** value. The HTML style attribute to  
2 obtain the **HtmlTextWriterStyle** value for.

### 3       GetStyleName

5 [C#]   protected   string   GetStyleName(HtmlTextWriterStyle   styleKey);

6 [C++]   protected:   String\*   GetStyleName(HtmlTextWriterStyle   styleKey);

7 [VB]   Protected   Function   GetStyleName(ByVal   styleKey   As  
8 HtmlTextWriterStyle)                               As                               String

9 [JScript] protected function GetStyleName(styleKey : HtmlTextWriterStyle) :  
10 String;

### 12       Description

13       Obtains the HTML style attribute associated with the specified  
14 **System.Web.UI.HtmlTextWriterStyle** enumeration value.

15 *Return Value:* The HTML style attribute. The **HtmlTextWriterStyle** value to  
16 obtain the HTML style attribute for.

### 17       GetTagKey

19 [C#]   protected   virtual   HtmlTextWriterTag   GetTagKey(string   tagName);

20 [C++]   protected:   virtual   HtmlTextWriterTag   GetTagKey(String\*   tagName);

21 [VB]   Overridable Protected Function GetTagKey(ByVal tagName As String) As  
22 HtmlTextWriterTag

23 [JScript] protected function GetTagKey(tagName : String) : HtmlTextWriterTag;

### 25       Description

Obtains the **System.Web.UI.HtmlTextWriterTag** enumeration value associated with the specified HTML element.

*Return Value:* The **HtmlTextWriterTag** value; if the *tagName* parameter is not associated with a specific **HtmlTextWriterTag** value, **HtmlTextWriterTag.Unknown** is returned. The HTML element to obtain the **HtmlTextWriterTag** value for.

#### GetTagName

[C#] protected virtual string GetTagName(HtmlTextWriterTag tagKey);

[C++] protected: virtual String\* GetTagName(HtmlTextWriterTag tagKey);

[VB] Overridable Protected Function GetTagName(ByVal tagKey As HtmlTextWriterTag) As String

[JScript] protected function GetTagName(tagKey : HtmlTextWriterTag) : String;

#### Description

Obtains the HTML element associated with the specified **System.Web.UI.HtmlTextWriterTag** enumeration value.

*Return Value:* The HTML element. The **HtmlTextWriterTag** value to obtain the HTML element for.

#### IsAttributeDefined

[C#] protected bool IsAttributeDefined(HtmlTextWriterAttribute key);

[C++] protected: bool IsAttributeDefined(HtmlTextWriterAttribute key);

[VB] Protected Function IsAttributeDefined(ByVal key As HtmlTextWriterAttribute) As Boolean

[JScript] protected function IsAttributeDefined(key : HtmlTextWriterAttribute) : Boolean; Determines whether an HTML attribute is recognized as one of the **System.Web.UI.HtmlTextWriterAttribute** values.

*Description*

Determines whether the specified HTML attribute is recognized.  
*Return Value:* **true** if the HTML attribute is associated with one of the **HtmlTextWriterAttribute** values; otherwise, **false**. The **System.Web.UI.HtmlTextWriterAttribute** associated with the HTML attribute.

**IsAttributeDefined**

[C#] protected bool IsAttributeDefined(HtmlTextWriterAttribute key, out string value);

[C++] protected: bool IsAttributeDefined(HtmlTextWriterAttribute key, String\*\* value);

[VB] Protected Function IsAttributeDefined(ByVal key As HtmlTextWriterAttribute, ByRef value As String) As Boolean

[JScript] protected function IsAttributeDefined(key : HtmlTextWriterAttribute, value : String) : Boolean;

*Description*

Determines whether the specified HTML attribute is recognized.  
*Return Value:* **true** if the HTML attribute is associated with one of the **HtmlTextWriterAttribute** values; otherwise, **false**. The

**System.Web.UI.HtmlTextWriterAttribute** associated with the HTML attribute.

The value assigned to the HTML attribute.

**IsStyleAttributeDefined**

[C#] protected bool IsStyleAttributeDefined(HtmlTextWriterStyle key);

[C++] protected: bool IsStyleAttributeDefined(HtmlTextWriterStyle key);

[VB] Protected Function IsStyleAttributeDefined(ByVal key As  
HtmlTextWriterStyle) As Boolean

[JScript] protected function IsStyleAttributeDefined(key : HtmlTextWriterStyle) :

Boolean; Determines whether an HTML style attribute is recognized as one of the

**System.Web.UI.HtmlTextWriterStyle** values.

### *Description*

Determines whether the specified HTML style attribute is recognized.

*Return Value:* **true** if the HTML attribute is associated with one of the

**HtmlTextWriterStyle** values; otherwise, **false**. The

**System.Web.UI.HtmlTextWriterStyle** value associated with the HTML  
attribute.

**IsStyleAttributeDefined**

[C#] protected bool IsStyleAttributeDefined(HtmlTextWriterStyle key, out string  
value);

[C++] protected: bool IsStyleAttributeDefined(HtmlTextWriterStyle key, String\*\*  
value);

[VB] Protected Function IsStyleAttributeDefined(ByVal key As

```

1 HtmlTextWriterStyle, ByRef value As String) As Boolean
2 [JScript] protected function IsStyleAttributeDefined(key : HtmlTextWriterStyle,
3 value : String) : Boolean;
4

```

#### 5 *Description*

6 Determines whether the specified HTML style attribute is recognized.

7 *Return Value:* **true** if the HTML attribute is associated with one of the

8 **HtmlTextWriterStyle** values; otherwise, **false**. The

9 **System.Web.UI.HtmlTextWriterStyle** value associated with the HTML

10 attribute. The value assigned to the HTML style attribute.

#### 11 *OnAttributeRender*

```

12
13 [C#] protected virtual bool OnAttributeRender(string name, string value,
14 HtmlTextWriterAttribute key);
15 [C++] protected: virtual bool OnAttributeRender(String* name, String* value,
16 HtmlTextWriterAttribute key);
17 [VB] Overridable Protected Function OnAttributeRender(ByVal name As String,
18 ByVal value As String, ByVal key As HtmlTextWriterAttribute) As Boolean
19 [JScript] protected function OnAttributeRender(name : String, value : String, key :
20 HtmlTextWriterAttribute) : Boolean;
21

```

#### 22 *Description*

23 Determines whether the specified HTML attribute and its value have been

24 rendered to the requesting page.

25 *Return Value:* **true** if the attribute has been rendered to the page; otherwise, **false**.



1 The HTML attribute to render. The value that is assigned to the HTML attribute.  
2 The **System.Web.UI.HtmlTextWriterAttribute** enumeration value associated  
3 with the HTML attribute.

#### 4 OnStyleAttributeRender

6 [C#] protected virtual bool OnStyleAttributeRender(string name, string value,  
7 HtmlTextWriterStyle key);

8 [C++] protected: virtual bool OnStyleAttributeRender(String\* name, String\*  
9 value, HtmlTextWriterStyle key);

10 [VB] Overridable Protected Function OnStyleAttributeRender(ByVal name As  
11 String, ByVal value As String, ByVal key As HtmlTextWriterStyle) As Boolean

12 [JScript] protected function OnStyleAttributeRender(name : String, value : String,  
13 key : HtmlTextWriterStyle : Boolean;

#### 15 Description

16 Determines whether the specified HTML style attribute and its value have  
17 been rendered to the requesting page.

18 *Return Value:* **true** if the HTML style attribute has been rendered to the page;  
19 otherwise, **false**. The HTML style attribute to render. The value that is assigned to  
20 the HTML style attribute. The **System.Web.UI.HtmlTextWriterStyle**  
21 enumeration value associated with the HTML style attribute.

#### 22 OnTagRender

24 [C#] protected virtual bool OnTagRender(string name, HtmlTextWriterTag key);

25 [C++] protected: virtual bool OnTagRender(String\* name, HtmlTextWriterTag

```

1 key);
2 [VB] Overridable Protected Function OnTagRender(ByVal name As String,
3 ByVal key As HtmlTextWriterTag) As Boolean
4 [JScript] protected function OnTagRender(name : String, key :
5 HtmlTextWriterTag) : Boolean;

```

#### 7 *Description*

8 Determines whether the specified HTML element has been rendered to the  
9 requesting page.

10 *Return Value:* **true** if the HTML element has been rendered to the page;  
11 otherwise, **false**. The HTML element to render. The  
12 **System.Web.UI.HtmlTextWriterTag** enumeration value associated with the  
13 HTML element.

#### 14 *OutputTabs*

```

15
16 [C#]          protected          virtual          void          OutputTabs();
17 [C++]          protected:          virtual          void          OutputTabs();
18 [VB]          Overridable          Protected          Sub          OutputTabs()
19 [JScript]          protected          function          OutputTabs();

```

#### 21 *Description*

22 Writes a series of blank characters that represent the tab spacing for a line  
23 of HTML characters.

This method uses the integer passed to the **System.Web.UI.HtmlTextWriter.Indent** property to determine how many spaces to write for the tab spacing.

#### PopEndTag

[C#]	protected	string	PopEndTag();
[C++]	protected:	String*	PopEndTag();
[VB]	Protected	Function	PopEndTag() As String
[JScript]	protected	function	PopEndTag() : String;

#### Description

Retrieves the appropriate closing tag for the HTML element to render to the page.

*Return Value:* The appropriate closing tag for the HTML element.

#### PushEndTag

[C#]	protected	void	PushEndTag(string endTag);
[C++]	protected:	void	PushEndTag(String* endTag);
[VB]	Protected	Sub	PushEndTag(ByVal endTag As String)
[JScript]	protected	function	PushEndTag(endTag : String);

#### Description

Associates the specified closing tag of an HTML element with the appropriate HTML text. The closing tag to be associated.

#### RegisterAttribute

1  
2 [C#] protected static void RegisterAttribute(string name, HtmlTextWriterAttribute  
3 key);

4 [C++] protected: static void RegisterAttribute(String\* name,  
5 HtmlTextWriterAttribute key);

6 [VB] Protected Shared Sub RegisterAttribute(ByVal name As String, ByVal key  
7 As HtmlTextWriterAttribute)

8 [JScript] protected static function RegisterAttribute(name : String, key :  
9 HtmlTextWriterAttribute);

10  
11 *Description*

12 Registers HTML attributes, whether literals or dynamically generated, from  
13 the source file so that they can be properly rendered to the requesting client. The  
14 HTML attribute to be registered. An **System.Web.UI.HtmlTextWriterAttribute**  
15 value that corresponds to the attribute name.

16 RegisterStyle

17  
18 [C#] protected static void RegisterStyle(string name, HtmlTextWriterStyle key);

19 [C++] protected: static void RegisterStyle(String\* name, HtmlTextWriterStyle  
20 key);

21 [VB] Protected Shared Sub RegisterStyle(ByVal name As String, ByVal key As  
22 HtmlTextWriterStyle)

23 [JScript] protected static function RegisterStyle(name : String, key :  
24 HtmlTextWriterStyle);

25

## *Description*

Registers HTML style properties, whether literals or dynamically generated, from the source file so that they can be properly rendered to the requesting client. The **System.String** passed from the source file specifying the style name. The **System.Web.UI.HtmlTextWriterStyle** value that corresponds to the specified style.

### RegisterTag

[C#] protected static void RegisterTag(string name, HtmlTextWriterTag key);

[C++] protected: static void RegisterTag(String\* name, HtmlTextWriterTag key);

[VB] Protected Shared Sub RegisterTag(ByVal name As String, ByVal key As HtmlTextWriterTag)

[JScript] protected static function RegisterTag(name : String, key : HtmlTextWriterTag);

## *Description*

Registers HTML tags, whether literals or dynamically generated, from the source file so that they can be properly rendered to the requesting client. A **System.String** that contains the HTML tag. An **System.Web.UI.HtmlTextWriterTag** value that specifies which element is to be rendered.

### RenderAfterContent

[C#] protected virtual string RenderAfterContent();

```

1 [C++]      protected:      virtual      String*      RenderAfterContent();
2 [VB]   Overridable   Protected   Function   RenderAfterContent()   As   String
3 [JScript]   protected   function   RenderAfterContent()   :   String;
4

```

#### 5 *Description*

6 Writes any text or spacing that occurs after the content of the HTML  
7 element to the **System.Web.UI.HtmlTextWriter** output stream.

8 *Return Value:* The spacing or text to write prior to the content of the HTML  
9 element. If not overridden, this method returns **null** .

#### 10 **RenderAfterTag**

```

11
12 [C#]      protected      virtual      string      RenderAfterTag();
13 [C++]      protected:      virtual      String*      RenderAfterTag();
14 [VB]   Overridable   Protected   Function   RenderAfterTag()   As   String
15 [JScript]   protected   function   RenderAfterTag()   :   String;
16

```

#### 17 *Description*

18 Writes any spacing or text that occurs after an HTML element's closing tag.

19 *Return Value:* The spacing or text to write after the closing tag of the HTML  
20 element. If not overridden, this method returns **null** .

#### 21 **RenderBeforeContent**

```

22
23 [C#]      protected      virtual      string      RenderBeforeContent();
24 [C++]      protected:      virtual      String*      RenderBeforeContent();
25 [VB]   Overridable   Protected   Function   RenderBeforeContent()   As   String

```

1 [JScript] protected function RenderBeforeContent() : String;

3 *Description*

4 Writes any tab spacing before the content contained in an HTML element.  
5 *Return Value:* The spacing to write prior to the content of the HTML element. If  
6 not overridden, this method returns **null**.

7 RenderBeforeTag

9 [C#] protected virtual string RenderBeforeTag();

10 [C++] protected: virtual String\* RenderBeforeTag();

11 [VB] Overridable Protected Function RenderBeforeTag() As String

12 [JScript] protected function RenderBeforeTag() : String;

14 *Description*

15 Writes any text or tab spacing that occurs before the opening tag of an  
16 HTML element to the **System.Web.UI.HtmlTextWriter** output stream.  
17 *Return Value:* The text or tab spacing to write to the output stream. If not  
18 overridden, this method returns **null**.

19 RenderBeginTag

21 [C#] public virtual void RenderBeginTag(HtmlTextWriterTag tagKey);

22 [C++] public: virtual void RenderBeginTag(HtmlTextWriterTag tagKey);

23 [VB] Overridable Public Sub RenderBeginTag(ByVal tagKey As  
24 HtmlTextWriterTag)

25 [JScript] public function RenderBeginTag(tagKey : HtmlTextWriterTag);

1  
2 *Description*

3 Writes the opening tag of the HTML element associated with the specified  
4 **System.Web.UI.HtmlTextWriterTag** enumeration value to the output stream.  
5 An **HtmlTextWriterTag** value that defines the opening tag of the HTML element  
6 to render.

7 **RenderBeginTag**

8  
9 [C#] public virtual void RenderBeginTag(string tagName);  
10 [C++] public: virtual void RenderBeginTag(String\* tagName);  
11 [VB] Overridable Public Sub RenderBeginTag(ByVal tagName As String)  
12 [JScript] public function RenderBeginTag(tagName : String); Writes the opening  
13 tag of an HTML element to the **System.Web.UI.HtmlTextWriter** output stream.  
14

15 *Description*

16 Writes the opening tag of the specified HTML element to the output  
17 stream. The HTML element to render the opening tag for.

18 **RenderEndTag**

19  
20 [C#] public virtual void RenderEndTag();  
21 [C++] public: virtual void RenderEndTag();  
22 [VB] Overridable Public Sub RenderEndTag()  
23 [JScript] public function RenderEndTag();  
24

25 *Description*



Writes the end tag of an HTML element to the **System.Web.UI.HtmlTextWriter** output stream.

Write

[C#] public override void Write(bool value);

[C++] public: void Write(bool value);

[VB] Overrides Public Sub Write(ByVal value As Boolean)

[JScript] public override function Write(value : Boolean);

#### *Description*

Writes the text representation of a **Boolean** value to the text stream, along with any specified tab spacing. The **Boolean** value to be written to the text stream.

Write

[C#] public override void Write(char value);

[C++] public: void Write(\_\_wchar\_t value);

[VB] Overrides Public Sub Write(ByVal value As Char)

[JScript] public override function Write(value : Char);

#### *Description*

Writes a unicode character to the text stream, along with any specified tab spacing. The unicode character to write to the text stream.

Write

[C#] public override void Write(char[] buffer);

```

1 [C++]      public:      void      Write(__wchar_t      buffer      __gc[]);
2 [VB]      Overrides      Public      Sub      Write(ByVal      buffer()      As      Char)
3 [JScript]      public      override      function      Write(buffer      :      Char[]);
4

```

#### *Description*

Writes a character array to the text stream, along with any specified tab spacing. The character array to write to the text stream.

#### *Write*

```

10 [C#]      public      override      void      Write(double      value);
11 [C++]      public:      void      Write(double      value);
12 [VB]      Overrides      Public      Sub      Write(ByVal      value      As      Double)
13 [JScript]      public      override      function      Write(value      :      double);
14

```

#### *Description*

Writes the text representation of a double-precision floating point number to the HTML text stream, along with any specified tab spacing. The double-precision floating point number to write to the text stream.

#### *Write*

```

21 [C#]      public      override      void      Write(int      value);
22 [C++]      public:      void      Write(int      value);
23 [VB]      Overrides      Public      Sub      Write(ByVal      value      As      Integer)
24 [JScript]      public      override      function      Write(value      :      int);
25

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

*Description*

Writes the text representation of a 32-byte signed integer to the text stream, along with any specified tab spacing. The 32-byte signed integer to write to the text stream.

Write

```
[C#]      public      override      void      Write(long      value);
[C++]      public:      void      Write(__int64      value);
[VB]      Overrides      Public      Sub      Write(ByVal      value      As      Long)
[JScript]      public      override      function      Write(value      :      long);
```

*Description*

Writes the text representation of an 64-byte signed integer to the text stream, along with any specified tab spacing. The 64-byte signed integer to write to the text stream.

Write

```
[C#]      public      override      void      Write(object      value);
[C++]      public:      void      Write(Object*      value);
[VB]      Overrides      Public      Sub      Write(ByVal      value      As      Object)
[JScript]      public      override      function      Write(value      :      Object);
```

*Description*

Writes the text representation of an **System.Object** to the text stream, along with any specified tab spacing. The **Object** to write to the text stream.

Write

[C#] public override void Write(float value);

[C++] public: void Write(float value);

[VB] Overrides Public Sub Write(ByVal value As Single)

[JScript] public override function Write(value : float);

#### *Description*

Writes the text representation of a single-precision floating point number to the HTML text stream, along with any specified tab spacing. The single-precision floating point number to write to the text stream.

Write

[C#] public override void Write(string s);

[C++] public: void Write(String\* s);

[VB] Overrides Public Sub Write(ByVal s As String)

[JScript] public override function Write(s : String); Writes the given data type to an HTML text stream to be rendered to an ASP.NET page.

#### *Description*

Writes the specified string to the text stream, along with any specified tab spacing. A **System.String** to be written to the text stream.

Write

```

1
2 [C#] public override void Write(string format, object arg0);
3 [C++] public: void Write(String* format, Object* arg0);
4 [VB] Overrides Public Sub Write(ByVal format As String, ByVal arg0 As Object)
5 [JScript] public override function Write(format : String, arg0 : Object);
6

```

### 7 *Description*

8 Writes a tab string and a formatted string to the HTML text stream, using  
9 the same semantics as **System.String.Format(System.String,System.Object)** .

10 The formatting string. An object to write into the formatted string.

### 11 Write

```

12
13 [C#] public override void Write(string format, params object[] arg);
14 [C++] public: void Write(String* format, Object* arg __gc[]);
15 [VB] Overrides Public Sub Write(ByVal format As String, ByVal ParamArray
16 arg() As Object)
17 [JScript] public override function Write(format : String, arg : Object[]);
18

```

### 19 *Description*

20 Writes a tab string and a formatted string to the HTML text stream, using  
21 the same semantics as **System.String.Format(System.String,System.Object)** .

22 The formatting string. The object array to write into the formatted string.

### 23 Write

```

24
25 [C#] public override void Write(char[] buffer, int index, int count);

```

```

1 [C++] public: void Write(__wchar_t buffer __gc[], int index, int count);
2 [VB] Overrides Public Sub Write(ByVal buffer() As Char, ByVal index As
3 Integer,          ByVal          count          As          Integer)
4 [JScript] public override function Write(buffer : Char[], index : int, count : int);
5

```

### *Description*

Writes a subarray of characters to the text stream, along with any specified tab spacing. The array of characters from which the subarray is written to the text stream. The index location in the array where writing begins. The number of characters to be written to the text stream.

### *Write*

```

13 [C#] public override void Write(string format, object arg0, object arg1);
14 [C++] public: void Write(String* format, Object* arg0, Object* arg1);
15 [VB] Overrides Public Sub Write(ByVal format As String, ByVal arg0 As Object,
16 ByVal          arg1          As          Object)
17 [JScript] public override function Write(format : String, arg0 : Object, arg1 :
18 Object);
19

```

### *Description*

Writes a tab string and a formatted string to the HTML text stream, using the same semantics as **System.String.Format(System.String,System.Object)** . The formatting string. An object to write into the formatted string. An object to write into the formatted string.

### *WriteAttribute*

1  
2 [C#] public virtual void WriteAttribute(string name, string value);  
3 [C++] public: virtual void WriteAttribute(String\* name, String\* value);  
4 [VB] Overridable Public Sub WriteAttribute(ByVal name As String, ByVal value  
5 As String)  
6 [JScript] public function WriteAttribute(name : String, value : String); Writes an  
7 HTML attribute and its value to the **System.Web.UI.HtmlTextWriter** output  
8 stream.

9  
10 *Description*

11 Writes the specified HTML attribute and value to the  
12 **System.Web.UI.HtmlTextWriter** output stream. The HTML attribute to write to  
13 the output stream. The value assigned to the HTML attribute.

14 **WriteAttribute**

15  
16 [C#] public virtual void WriteAttribute(string name, string value, bool fEncode);  
17 [C++] public: virtual void WriteAttribute(String\* name, String\* value, bool  
18 fEncode);  
19 [VB] Overridable Public Sub WriteAttribute(ByVal name As String, ByVal value  
20 As String, ByVal fEncode As Boolean)  
21 [JScript] public function WriteAttribute(name : String, value : String, fEncode :  
22 Boolean);

23  
24 *Description*

25

Writes the specified HTML attribute and value to the **System.Web.UI.HtmlTextWriter** output stream and HTML encodes them if specified in the *fEncode* parameter. The HTML attribute to write to the output stream. The value assigned to the HTML attribute. **true** to HTML encode the attribute and its assigned value; otherwise, **false**.

#### WriteBeginTag

```
[C#]      public      virtual      void      WriteBeginTag(string      tagName);
[C++]     public:     virtual      void      WriteBeginTag(String*      tagName);
[VB]      Overridable Public Sub WriteBeginTag(ByVal tagName As String)
[JScript] public      function      WriteBeginTag(tagName      :      String);
```

#### Description

Writes any tab spacing and the opening tag of the specified HTML element to the **System.Web.UI.HtmlTextWriter** output stream.

This method does not write the closing character (>) of the HTML element's opening tag. Use this method with the **System.Web.UI.HtmlTextWriter.SelfClosingTagEnd** constant when you write HTML elements that are self closing. The HTML element to write the opening tag of.

#### WriteEndTag

```
[C#]      public      virtual      void      WriteEndTag(string      tagName);
[C++]     public:     virtual      void      WriteEndTag(String*      tagName);
[VB]      Overridable Public Sub WriteEndTag(ByVal tagName As String)
```



1 [JScript] public function WriteEndTag(tagName : String);

3 *Description*

4 Writes any tab spacing and the closing tag of the specified HTML element.  
5 The HTML element to write the closing tag for.

6 WriteFullBeginTag

8 [C#] public virtual void WriteFullBeginTag(string tagName);

9 [C++] public: virtual void WriteFullBeginTag(String\* tagName);

10 [VB] Overridable Public Sub WriteFullBeginTag(ByVal tagName As String)

11 [JScript] public function WriteFullBeginTag(tagName : String);

13 *Description*

14 Writes any tab spacing and the opening tag of the specified HTML element  
15 to the **System.Web.UI.HtmlTextWriter** output stream.

16 This method automatically writes the closing character (>) of the opening  
17 tag of the HTML element. The HTML element to write.

18 WriteLine

20 [C#] public override void WriteLine();

21 [C++] public: void WriteLine();

22 [VB] Overrides Public Sub WriteLine()

23 [JScript] public override function WriteLine();

25 *Description*

Writes a line terminator to the HTML text stream.

WriteLine

[C#] public override void WriteLine(bool value);

[C++] public: void WriteLine(bool value);

[VB] Overrides Public Sub WriteLine(ByVal value As Boolean)

[JScript] public override function WriteLine(value : Boolean);

### *Description*

Writes a tab string and the text representation of a **Boolean** followed by a line terminator to the HTML text stream. The **Boolean** to be written to the text stream.

WriteLine

[C#] public override void WriteLine(char value);

[C++] public: void WriteLine(\_\_wchar\_t value);

[VB] Overrides Public Sub WriteLine(ByVal value As Char)

[JScript] public override function WriteLine(value : Char);

### *Description*

Writes a tab string and a character followed by a line terminator to the HTML text stream. The character to be written to the text stream.

WriteLine

[C#] public override void WriteLine(char[] buffer);

```

1 [C++]      public:      void      WriteLine(__wchar_t      buffer      __gc[]);
2 [VB]      Overrides      Public      Sub      WriteLine(ByVal      buffer()      As      Char)
3 [JScript]      public      override      function      WriteLine(buffer      :      Char[]);
4

```

#### 5 *Description*

6 Writes a tab string and a character array followed by a line terminator to the  
7 HTML text stream. The character array to be written to the text stream.

8 WriteLine

```

9
10 [C#]      public      override      void      WriteLine(double      value);
11 [C++]      public:      void      WriteLine(double      value);
12 [VB]      Overrides      Public      Sub      WriteLine(ByVal      value      As      Double)
13 [JScript]      public      override      function      WriteLine(value      :      double);
14

```

#### 15 *Description*

16 Writes a tab string and the text representation of an 8-byte floating-point  
17 value, followed by a line terminator, to the HTML text stream. The 8-byte  
18 floating-point value to write to the text stream.

19 WriteLine

```

20
21 [C#]      public      override      void      WriteLine(int      value);
22 [C++]      public:      void      WriteLine(int      value);
23 [VB]      Overrides      Public      Sub      WriteLine(ByVal      value      As      Integer)
24 [JScript]      public      override      function      WriteLine(value      :      int);
25

```

1  
2 *Description*

3 Writes a tab string and the text representation of an 32-byte signed integer,  
4 followed by a line terminator, to the HTML text stream. The 32-byte signed  
5 integer to write to the text stream.

6 WriteLine

7  
8 [C#] public override void WriteLine(long value);  
9 [C++] public: void WriteLine(\_\_int64 value);  
10 [VB] Overrides Public Sub WriteLine(ByVal value As Long)  
11 [JScript] public override function WriteLine(value : long);  
12

13 *Description*

14 Writes a tab string and the text representation of an 64-byte signed integer,  
15 followed by a line terminator, to the HTML text stream. The 64-byte signed  
16 integer to write to the text stream.

17 WriteLine

18  
19 [C#] public override void WriteLine(object value);  
20 [C++] public: void WriteLine(Object\* value);  
21 [VB] Overrides Public Sub WriteLine(ByVal value As Object)  
22 [JScript] public override function WriteLine(value : Object);  
23

24 *Description*  
25

Writes any tab strings and the text representation of an **System.Object** , followed by a line terminator, to the HTML text stream. The **Object** to write to the text stream.

#### WriteLine

[C#] public override void WriteLine(float value);

[C++] public: void WriteLine(float value);

[VB] Overrides Public Sub WriteLine(ByVal value As Single)

[JScript] public override function WriteLine(value : float);

#### *Description*

Writes a tab string and the text representation of a single-precision floating point number, followed by a line terminator, to the HTML text stream. The single-precision floating point number to write to the text string.

#### WriteLine

[C#] public override void WriteLine(string s);

[C++] public: void WriteLine(String\* s);

[VB] Overrides Public Sub WriteLine(ByVal s As String)

[JScript] public override function WriteLine(s : String); Writes some data to an HTML text stream as specified by the overloaded parameters, followed by a line terminator. All instances of this method write tab strings to the text stream.

#### *Description*

Writes a tab string and a **System.String** followed by a line terminator to an HTML text stream. The **String** to write to the text stream.

### WriteLine

```
[C#]      public      override      void      WriteLine(uint      value);  
[C++]     public:     void      WriteLine(unsigned      int      value);  
[VB]      Overrides   Public   Sub   WriteLine(ByVal   value   As   UInt32)  
[JScript] public      override      function      WriteLine(value      :      UInt32);
```

### *Description*

Writes any tab strings and the text representation of a 4-byte unsigned integer followed by a line terminator to the HTML text stream. The 4-byte unsigned integer to write.

### WriteLine

```
[C#]      public      override      void      WriteLine(string      format,      object      arg0);  
[C++]     public:     void      WriteLine(String*      format,      Object*      arg0);  
[VB]      Overrides   Public   Sub   WriteLine(ByVal   format   As   String,   ByVal   arg0   As  
Object)  
[JScript] public      override      function      WriteLine(format      :      String,      arg0      :      Object);
```

### *Description*

Writes any tab strings and a formatted string, followed by a line terminator, to the HTML text stream. The method uses the same semantics as

1 **System.String.Format(System.String,System.Object)** . The formatting string.  
2 The object to write into the formatted string.

3 WriteLine

4  
5 [C#] public override void WriteLine(string format, params object[] arg);  
6 [C++] public: void WriteLine(String\* format, Object\* arg \_\_gc[]);  
7 [VB] Overrides Public Sub WriteLine(ByVal format As String, ByVal  
8 ParamArray arg() As Object)  
9 [JScript] public override function WriteLine(format : String, arg : Object[]);  
10

11 *Description*

12 Writes any tab strings and a formatted string, followed by a line terminator,  
13 to the HTML text stream. The method uses the same semantics as  
14 **System.String.Format(System.String,System.Object)** . The formatting string.  
15 The array of objects to write into the formatted string.

16 WriteLine

17  
18 [C#] public override void WriteLine(char[] buffer, int index, int count);  
19 [C++] public: void WriteLine(\_\_wchar\_t buffer \_\_gc[], int index, int count);  
20 [VB] Overrides Public Sub WriteLine(ByVal buffer() As Char, ByVal index As  
21 Integer, ByVal count As Integer)  
22 [JScript] public override function WriteLine(buffer : Char[], index : int, count :  
23 int);  
24

25 *Description*

Writes a tab string and a subarray of characters followed by a line terminator to the HTML text stream. The character array from which to write to the text stream. The location in the character array where writing begins. The number of characters in the array to write to the text stream.

#### WriteLine

```
[C#] public override void WriteLine(string format, object arg0, object arg1);
[C++] public: void WriteLine(String* format, Object* arg0, Object* arg1);
[VB] Overrides Public Sub WriteLine(ByVal format As String, ByVal arg0 As
Object,          ByVal          arg1          As          Object)
[JScript] public override function WriteLine(format : String, arg0 : Object, arg1 :
Object);
```

#### *Description*

Writes any tab strings and a formatted string, followed by a line terminator, to the HTML text stream. The method uses the same semantics as **System.String.Format(System.String,System.Object)** . The formatting string. An object to write into the formatted string. An object to write into the formatted string.

#### WriteLineNoTabs

```
[C#]      public      void      WriteLineNoTabs(string      s);
[C++]      public:      void      WriteLineNoTabs(String*      s);
[VB]      Public      Sub      WriteLineNoTabs(ByVal      s      As      String)
[JScript]      public      function      WriteLineNoTabs(s      :      String);
```



1  
2 *Description*

3 Writes a **System.String** followed by a line terminator to an HTML text  
4 stream. This method ignores any specified tab spacing. The **String** to write to the  
5 HTML text stream.

6 **WriteStyleAttribute**

7  
8 [C#] public virtual void WriteStyleAttribute(string name, string value);

9 [C++] public: virtual void WriteStyleAttribute(String\* name, String\* value);

10 [VB] Overridable Public Sub WriteStyleAttribute(ByVal name As String, ByVal  
11 value As String)

12 [JScript] public function WriteStyleAttribute(name : String, value : String); Writes  
13 an HTML style attribute and its value to the **System.Web.UI.HtmlTextWriter**  
14 output stream.

15  
16 *Description*

17 Writes the specified HTML style attribute to the  
18 **System.Web.UI.HtmlTextWriter** output stream. The HTML style attribute to  
19 write to the output stream. The value assigned to the HTML style attribute.

20 **WriteStyleAttribute**

21  
22 [C#] public virtual void WriteStyleAttribute(string name, string value, bool  
23 fEncode);

24 [C++] public: virtual void WriteStyleAttribute(String\* name, String\* value, bool  
25 fEncode);

```

1 [VB] Overridable Public Sub WriteStyleAttribute(ByVal name As String, ByVal
2 value As String, ByVal fEncode As Boolean)
3 [JScript] public function WriteStyleAttribute(name : String, value : String,
4 fEncode : Boolean);

```

#### 6 *Description*

7 Writes the specified HTML attribute and value to the  
8 **System.Web.UI.HtmlTextWriter** output stream and HTML encodes them if  
9 specified in the *fEncode* parameter. The HTML style attribute to write to the  
10 output stream. The value assigned to the HTML style attribute. **true** to HTML  
11 encode the style attribute and its assigned value; otherwise, **false**.

12 HtmlTextWriterAttribute enumeration (System.Web.UI)

13 WriteStyleAttribute

#### 16 *Description*

17 Specifies the HTML attributes that an **System.Web.UI.HtmlTextWriter**  
18 or **System.Web.UI.Html32TextWriter** object writes to the opening tag of an  
19 HTML element when a Web request is processed.

20 WriteStyleAttribute

```

22 [C#] public const HtmlTextWriterAttribute Accesskey;
23 [C++] public: const HtmlTextWriterAttribute Accesskey;
24 [VB] Public Const Accesskey As HtmlTextWriterAttribute
25 [JScript] public var Accesskey : HtmlTextWriterAttribute;

```

1  
2 *Description*

3 Specifies that the HTML **accesskey** attribute should be written to the tag.

4 WriteStyleAttribute

5  
6 [C#] public const HtmlTextWriterAttribute Align;

7 [C++] public: const HtmlTextWriterAttribute Align;

8 [VB] Public Const Align As HtmlTextWriterAttribute

9 [JScript] public var Align : HtmlTextWriterAttribute;

10  
11 *Description*

12 Specifies that the HTML **align** attribute should be written to the tag.

13 WriteStyleAttribute

14  
15 [C#] public const HtmlTextWriterAttribute Alt;

16 [C++] public: const HtmlTextWriterAttribute Alt;

17 [VB] Public Const Alt As HtmlTextWriterAttribute

18 [JScript] public var Alt : HtmlTextWriterAttribute;

19  
20 *Description*

21 Specifies that the HTML **alt** attribute should be written to the tag.

22 WriteStyleAttribute

23  
24 [C#] public const HtmlTextWriterAttribute Background;

25 [C++] public: const HtmlTextWriterAttribute Background;

```

1  [VB]      Public      Const      Background      As      HtmlTextWriterAttribute
2  [JScript] public      var      Background      :      HtmlTextWriterAttribute;

```

#### *Description*

Specifies that the HTML **background** attribute should be written to the tag.

WriteStyleAttribute

```

8  [C#]      public      const      HtmlTextWriterAttribute      Bgcolor;
9  [C++]     public:     const      HtmlTextWriterAttribute      Bgcolor;
10 [VB]      Public      Const      Bgcolor      As      HtmlTextWriterAttribute
11 [JScript] public      var      Bgcolor      :      HtmlTextWriterAttribute;

```

#### *Description*

Specifies that the HTML **bgcolor** attribute should be written to the tag.

WriteStyleAttribute

```

17 [C#]      public      const      HtmlTextWriterAttribute      Border;
18 [C++]     public:     const      HtmlTextWriterAttribute      Border;
19 [VB]      Public      Const      Border      As      HtmlTextWriterAttribute
20 [JScript] public      var      Border      :      HtmlTextWriterAttribute;

```

#### *Description*

Specifies that the HTML **border** attribute should be written to the tag.

WriteStyleAttribute

```

1
2 [C#]      public      const      HtmlTextWriterAttribute      Bordercolor;
3 [C++]     public:     const      HtmlTextWriterAttribute      Bordercolor;
4 [VB]      Public      Const      Bordercolor      As      HtmlTextWriterAttribute
5 [JScript] public      var      Bordercolor      :      HtmlTextWriterAttribute;
6

```

#### 7 *Description*

8 Specifies that the HTML **bordercolor** attribute should be written to the tag.

9 WriteStyleAttribute

```

10
11 [C#]      public      const      HtmlTextWriterAttribute      Cellpadding;
12 [C++]     public:     const      HtmlTextWriterAttribute      Cellpadding;
13 [VB]      Public      Const      Cellpadding      As      HtmlTextWriterAttribute
14 [JScript] public      var      Cellpadding      :      HtmlTextWriterAttribute;
15

```

#### 16 *Description*

17 Specifies that the HTML **cellpadding** attribute should be written to the tag.

18 WriteStyleAttribute

```

19
20 [C#]      public      const      HtmlTextWriterAttribute      Cellspacing;
21 [C++]     public:     const      HtmlTextWriterAttribute      Cellspacing;
22 [VB]      Public      Const      Cellspacing      As      HtmlTextWriterAttribute
23 [JScript] public      var      Cellspacing      :      HtmlTextWriterAttribute;
24

```

#### 25 *Description*

Specifies that the HTML **cellspacing** attribute should be written to the tag.

WriteStyleAttribute

[C#] public const HtmlTextWriterAttribute Checked;

[C++] public: const HtmlTextWriterAttribute Checked;

[VB] Public Const Checked As HtmlTextWriterAttribute

[JScript] public var Checked : HtmlTextWriterAttribute;

*Description*

Specifies that the HTML **checked** attribute should be written to the tag.

WriteStyleAttribute

[C#] public const HtmlTextWriterAttribute Class;

[C++] public: const HtmlTextWriterAttribute Class;

[VB] Public Const Class As HtmlTextWriterAttribute

[JScript] public var Class : HtmlTextWriterAttribute;

*Description*

Specifies that the HTML **class** attribute should be written to the tag.

WriteStyleAttribute

[C#] public const HtmlTextWriterAttribute Cols;

[C++] public: const HtmlTextWriterAttribute Cols;

[VB] Public Const Cols As HtmlTextWriterAttribute

[JScript] public var Cols : HtmlTextWriterAttribute;

*Description*

Specifies that the HTML **cols** attribute should be written to the tag.

WriteStyleAttribute

[C#]        public        const        HtmlTextWriterAttribute        Colspan;

[C++]       public:        const        HtmlTextWriterAttribute        Colspan;

[VB]        Public        Const        Colspan        As        HtmlTextWriterAttribute

[JScript]    public        var        Colspan        :        HtmlTextWriterAttribute;

*Description*

Specifies that the HTML **colspan** attribute should be written to the tag.

WriteStyleAttribute

[C#]        public        const        HtmlTextWriterAttribute        Disabled;

[C++]       public:        const        HtmlTextWriterAttribute        Disabled;

[VB]        Public        Const        Disabled        As        HtmlTextWriterAttribute

[JScript]    public        var        Disabled        :        HtmlTextWriterAttribute;

*Description*

Specifies that the HTML **disabled** attribute should be written to the tag.

WriteStyleAttribute

[C#]        public        const        HtmlTextWriterAttribute        For;

[C++]       public:        const        HtmlTextWriterAttribute        For;

1	[VB]	Public	Const	For	As	HtmlTextWriterAttribute
2	[JScript]	public	var	For	:	HtmlTextWriterAttribute;

3

4 *Description*

5 Specifies that the HTML **for** attribute should be written to the tag.

6 WriteStyleAttribute

8	[C#]	public	const	HtmlTextWriterAttribute	Height;
9	[C++]	public:	const	HtmlTextWriterAttribute	Height;
10	[VB]	Public	Const	Height	As HtmlTextWriterAttribute
11	[JScript]	public	var	Height	: HtmlTextWriterAttribute;

12

13 *Description*

14 Specifies that the HTML **height** attribute should be written to the tag.

15 WriteStyleAttribute

17	[C#]	public	const	HtmlTextWriterAttribute	Href;
18	[C++]	public:	const	HtmlTextWriterAttribute	Href;
19	[VB]	Public	Const	Href	As HtmlTextWriterAttribute
20	[JScript]	public	var	Href	: HtmlTextWriterAttribute;

21

22 *Description*

23 Specifies that the HTML **href** attribute should be written to the tag.

24 WriteStyleAttribute

25



1						
2	[C#]	public	const		HtmlTextWriterAttribute	Id;
3	[C++]	public:	const		HtmlTextWriterAttribute	Id;
4	[VB]	Public	Const	Id	As	HtmlTextWriterAttribute
5	[JScript]	public	var	Id	:	HtmlTextWriterAttribute;

*Description*

Specifies that the HTML **id** attribute should be written to the tag.

WriteStyleAttribute

11	[C#]	public	const		HtmlTextWriterAttribute	Maxlength;
12	[C++]	public:	const		HtmlTextWriterAttribute	Maxlength;
13	[VB]	Public	Const	Maxlength	As	HtmlTextWriterAttribute
14	[JScript]	public	var	Maxlength	:	HtmlTextWriterAttribute;

*Description*

Specifies that the HTML **maxlength** attribute should be written to the tag.

WriteStyleAttribute

20	[C#]	public	const		HtmlTextWriterAttribute	Multiple;
21	[C++]	public:	const		HtmlTextWriterAttribute	Multiple;
22	[VB]	Public	Const	Multiple	As	HtmlTextWriterAttribute
23	[JScript]	public	var	Multiple	:	HtmlTextWriterAttribute;

*Description*



*Description*

Specifies that the HTML **onchange** attribute should be written to the tag.

WriteStyleAttribute

[C#]        public        const        HtmlTextWriterAttribute        Onclick;

[C++]       public:        const        HtmlTextWriterAttribute        Onclick;

[VB]        Public        Const        Onclick        As        HtmlTextWriterAttribute

[JScript]    public        var        Onclick        :        HtmlTextWriterAttribute;

*Description*

Specifies that the HTML **onclick** attribute should be written to the tag.

WriteStyleAttribute

[C#]        public        const        HtmlTextWriterAttribute        ReadOnly;

[C++]       public:        const        HtmlTextWriterAttribute        ReadOnly;

[VB]        Public        Const        ReadOnly        As        HtmlTextWriterAttribute

[JScript]    public        var        ReadOnly        :        HtmlTextWriterAttribute;

*Description*

Specifies that the HTML **readonly** attribute should be written to the tag.

WriteStyleAttribute

[C#]        public        const        HtmlTextWriterAttribute        Rows;

[C++]       public:        const        HtmlTextWriterAttribute        Rows;

1 [VB] Public Const Rows As HtmlTextWriterAttribute  
 2 [JScript] public var Rows : HtmlTextWriterAttribute;

3  
 4 *Description*

5 Specifies that the HTML **rows** attribute should be written to the tag.  
 6 WriteStyleAttribute

7  
 8 [C#] public const HtmlTextWriterAttribute Rowspan;  
 9 [C++] public: const HtmlTextWriterAttribute Rowspan;  
 10 [VB] Public Const Rowspan As HtmlTextWriterAttribute  
 11 [JScript] public var Rowspan : HtmlTextWriterAttribute;

12  
 13 *Description*

14 Specifies that the HTML **rowspan** attribute should be written to the tag.  
 15 WriteStyleAttribute

16  
 17 [C#] public const HtmlTextWriterAttribute Rules;  
 18 [C++] public: const HtmlTextWriterAttribute Rules;  
 19 [VB] Public Const Rules As HtmlTextWriterAttribute  
 20 [JScript] public var Rules : HtmlTextWriterAttribute;

21  
 22 *Description*

23 Specifies that the HTML **rules** attribute should be written to the tag.  
 24 WriteStyleAttribute

25

1						
2	[C#]	public	const	HtmlTextWriterAttribute	Selected;	
3	[C++]	public:	const	HtmlTextWriterAttribute	Selected;	
4	[VB]	Public	Const	Selected	As	HtmlTextWriterAttribute
5	[JScript]	public	var	Selected	:	HtmlTextWriterAttribute;

6

7 *Description*

8 Specifies that the HTML **selected** attribute should be written to the tag.

9 WriteStyleAttribute

10						
11	[C#]	public	const	HtmlTextWriterAttribute	Size;	
12	[C++]	public:	const	HtmlTextWriterAttribute	Size;	
13	[VB]	Public	Const	Size	As	HtmlTextWriterAttribute
14	[JScript]	public	var	Size	:	HtmlTextWriterAttribute;

15

16 *Description*

17 Specifies that the HTML **size** attribute should be written to the tag.

18 WriteStyleAttribute

19						
20	[C#]	public	const	HtmlTextWriterAttribute	Src;	
21	[C++]	public:	const	HtmlTextWriterAttribute	Src;	
22	[VB]	Public	Const	Src	As	HtmlTextWriterAttribute
23	[JScript]	public	var	Src	:	HtmlTextWriterAttribute;

24

25 *Description*

Specifies that the HTML **src** attribute should be written to the tag.

WriteStyleAttribute

[C#]            public            const            HtmlTextWriterAttribute            Style;

[C++]            public:            const            HtmlTextWriterAttribute            Style;

[VB]            Public            Const            Style            As            HtmlTextWriterAttribute

[JScript]            public            var            Style            :            HtmlTextWriterAttribute;

*Description*

Specifies that the HTML **style** attribute should be written to the tag.

WriteStyleAttribute

[C#]            public            const            HtmlTextWriterAttribute            Tabindex;

[C++]            public:            const            HtmlTextWriterAttribute            Tabindex;

[VB]            Public            Const            Tabindex            As            HtmlTextWriterAttribute

[JScript]            public            var            Tabindex            :            HtmlTextWriterAttribute;

*Description*

Specifies that the HTML **tabindex** attribute should be written to the tag.

WriteStyleAttribute

[C#]            public            const            HtmlTextWriterAttribute            Target;

[C++]            public:            const            HtmlTextWriterAttribute            Target;

[VB]            Public            Const            Target            As            HtmlTextWriterAttribute

[JScript]            public            var            Target            :            HtmlTextWriterAttribute;

*Description*

Specifies that the HTML **target** attribute should be written to the tag.

WriteStyleAttribute

[C#]        public        const        HtmlTextWriterAttribute        Title;

[C++]        public:        const        HtmlTextWriterAttribute        Title;

[VB]        Public        Const        Title        As        HtmlTextWriterAttribute

[JScript]        public        var        Title        :        HtmlTextWriterAttribute;

*Description*

Specifies that the HTML **title** attribute should be written to the tag.

WriteStyleAttribute

[C#]        public        const        HtmlTextWriterAttribute        Type;

[C++]        public:        const        HtmlTextWriterAttribute        Type;

[VB]        Public        Const        Type        As        HtmlTextWriterAttribute

[JScript]        public        var        Type        :        HtmlTextWriterAttribute;

*Description*

Specifies that the HTML **type** attribute should be written to the tag.

WriteStyleAttribute

[C#]        public        const        HtmlTextWriterAttribute        Valign;

[C++]        public:        const        HtmlTextWriterAttribute        Valign;

```

1  [VB]      Public      Const      Valign      As      HtmlTextWriterAttribute
2  [JScript] public      var      Valign      :      HtmlTextWriterAttribute;

```

```

3
4  Description

```

```

5      Specifies that the HTML valign attribute should be written to the tag.
6      WriteStyleAttribute

```

```

7
8  [C#]      public      const      HtmlTextWriterAttribute      Value;
9  [C++]     public:      const      HtmlTextWriterAttribute      Value;
10 [VB]      Public      Const      Value      As      HtmlTextWriterAttribute
11 [JScript] public      var      Value      :      HtmlTextWriterAttribute;

```

```

12
13 Description

```

```

14      Specifies that the HTML value attribute should be written to the tag.
15      WriteStyleAttribute

```

```

16
17 [C#]      public      const      HtmlTextWriterAttribute      Width;
18 [C++]     public:      const      HtmlTextWriterAttribute      Width;
19 [VB]      Public      Const      Width      As      HtmlTextWriterAttribute
20 [JScript] public      var      Width      :      HtmlTextWriterAttribute;

```

```

21
22 Description

```

```

23      Specifies that the HTML width attribute should be written to the tag.
24      WriteStyleAttribute

```

```

25

```



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#]      public      const      HtmlTextWriterAttribute      Wrap;  
[C++]     public:     const      HtmlTextWriterAttribute      Wrap;  
[VB]      Public      Const      Wrap      As      HtmlTextWriterAttribute  
[JScript] public      var      Wrap      :      HtmlTextWriterAttribute;
```

*Description*

Specifies that the HTML **wrap** attribute should be written to the tag.  
HtmlTextWriterStyle enumeration (System.Web.UI)  
ToString

*Description*

Specifies the HTML styles available to an  
**System.Web.UI.HtmlTextWriter** or **System.Web.UI.Html32TextWriter** object  
output stream.  
**HtmlTextWriter** and **Html32TextWriter** objects use these enumeration  
values to register HTML strings to the proper HTML style.  
ToString

```
[C#]      public      const      HtmlTextWriterStyle      BackgroundColor;  
[C++]     public:     const      HtmlTextWriterStyle      BackgroundColor;  
[VB]      Public      Const      BackgroundColor      As      HtmlTextWriterStyle  
[JScript] public      var      BackgroundColor      :      HtmlTextWriterStyle;
```

1  
2 *Description*

3 Specifies the HTML **backgroundcolor** style.

4 ToString

5  
6 [C#] public const HtmlTextWriterStyle BackgroundImage;

7 [C++] public: const HtmlTextWriterStyle BackgroundImage;

8 [VB] Public Const BackgroundImage As HtmlTextWriterStyle

9 [JScript] public var BackgroundImage : HtmlTextWriterStyle;

10  
11 *Description*

12 Specifies the HTML **backgroundimage** style.

13 ToString

14  
15 [C#] public const HtmlTextWriterStyle BorderCollapse;

16 [C++] public: const HtmlTextWriterStyle BorderCollapse;

17 [VB] Public Const BorderCollapse As HtmlTextWriterStyle

18 [JScript] public var BorderCollapse : HtmlTextWriterStyle;

19  
20 *Description*

21 Specifies the HTML **bordercollapse** style.

22 ToString

23  
24 [C#] public const HtmlTextWriterStyle BorderColor;

25 [C++] public: const HtmlTextWriterStyle BorderColor;

```

1  [VB]      Public      Const      BorderColor      As      HtmlTextWriterStyle
2  [JScript] public      var      BorderColor      :      HtmlTextWriterStyle;

```

```

3
4  Description

```

```

5      Specifies the HTML bordercolor style.
6      ToString

```

```

8  [C#]      public      const      HtmlTextWriterStyle      BorderStyle;
9  [C++]      public:      const      HtmlTextWriterStyle      BorderStyle;
10 [VB]      Public      Const      BorderStyle      As      HtmlTextWriterStyle
11 [JScript] public      var      BorderStyle      :      HtmlTextWriterStyle;

```

```

12
13 Description
14      Specifies the HTML borderstyle style.
15      ToString

```

```

17 [C#]      public      const      HtmlTextWriterStyle      BorderWidth;
18 [C++]      public:      const      HtmlTextWriterStyle      BorderWidth;
19 [VB]      Public      Const      BorderWidth      As      HtmlTextWriterStyle
20 [JScript] public      var      BorderWidth      :      HtmlTextWriterStyle;

```

```

21
22 Description
23      Specifies the HTML borderwidth style.
24      ToString

```

```

25

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#]      public      const      HtmlTextWriterStyle      Color;  
[C++]     public:     const      HtmlTextWriterStyle      Color;  
[VB]      Public      Const      Color      As      HtmlTextWriterStyle  
[JScript] public      var      Color      :      HtmlTextWriterStyle;
```

*Description*

Specifies the HTML **color** style.

ToString

```
[C#]      public      const      HtmlTextWriterStyle      FontFamily;  
[C++]     public:     const      HtmlTextWriterStyle      FontFamily;  
[VB]      Public      Const      FontFamily      As      HtmlTextWriterStyle  
[JScript] public      var      FontFamily      :      HtmlTextWriterStyle;
```

*Description*

Specifies the HTML **fontfamily** style.

ToString

```
[C#]      public      const      HtmlTextWriterStyle      FontSize;  
[C++]     public:     const      HtmlTextWriterStyle      FontSize;  
[VB]      Public      Const      FontSize      As      HtmlTextWriterStyle  
[JScript] public      var      FontSize      :      HtmlTextWriterStyle;
```

*Description*

Specifies the HTML **fontsize** style.

ToString

[C#]            public            const            HtmlTextWriterStyle            FontStyle;

[C++]           public:            const            HtmlTextWriterStyle            FontStyle;

[VB]           Public           Const           FontStyle           As           HtmlTextWriterStyle

[JScript]       public           var           FontStyle           :           HtmlTextWriterStyle;

*Description*

Specifies the HTML **fontstyle** style.

ToString

[C#]            public            const            HtmlTextWriterStyle            FontWeight;

[C++]           public:            const            HtmlTextWriterStyle            FontWeight;

[VB]           Public           Const           FontWeight           As           HtmlTextWriterStyle

[JScript]       public           var           FontWeight           :           HtmlTextWriterStyle;

*Description*

Specifies the HTML **fontweight** style.

ToString

[C#]            public            const            HtmlTextWriterStyle            Height;

[C++]           public:            const            HtmlTextWriterStyle            Height;

[VB]           Public           Const           Height           As           HtmlTextWriterStyle

[JScript]       public           var           Height           :           HtmlTextWriterStyle;

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

*Description*

Specifies the HTML **height** style.

ToString

```
[C#]      public      const      HtmlTextWriterStyle      TextDecoration;
[C++]     public:     const      HtmlTextWriterStyle      TextDecoration;
[VB]      Public      Const      TextDecoration      As      HtmlTextWriterStyle
[JScript] public      var      TextDecoration      :      HtmlTextWriterStyle;
```

*Description*

Specifies the HTML **textdecoration** style.

ToString

```
[C#]      public      const      HtmlTextWriterStyle      Width;
[C++]     public:     const      HtmlTextWriterStyle      Width;
[VB]      Public      Const      Width      As      HtmlTextWriterStyle
[JScript] public      var      Width      :      HtmlTextWriterStyle;
```

*Description*

Specifies the HTML **width** style.

HtmlTextWriterTag enumeration (System.Web.UI)

ToString

1  
2  
3 *Description*

4 Specifies the HTML tags that can be passed to an  
5 **System.Web.UI.HtmlTextWriter** or **System.Web.UI.Html32TextWriter** object  
6 output stream.

7 This enumeration allows the output stream to write HTML mark up, along  
8 with HTML server controls, in response to a Web request.

9 *ToString*

10  
11 [C#] public const HtmlTextWriterTag A;  
12 [C++] public: const HtmlTextWriterTag A;  
13 [VB] Public Const A As HtmlTextWriterTag  
14 [JScript] public var A : HtmlTextWriterTag;  
15

16 *Description*

17 Specifies the HTML a element.

18 *ToString*

19  
20 [C#] public const HtmlTextWriterTag Acronym;  
21 [C++] public: const HtmlTextWriterTag Acronym;  
22 [VB] Public Const Acronym As HtmlTextWriterTag  
23 [JScript] public var Acronym : HtmlTextWriterTag;  
24

25 *Description*

1 Specifies the HTML **acronym** element.

2 ToString

3

4 [C#] public const HtmlTextWriterTag Address;

5 [C++] public: const HtmlTextWriterTag Address;

6 [VB] Public Const Address As HtmlTextWriterTag

7 [JScript] public var Address : HtmlTextWriterTag;

8

9 *Description*

10 Specifies the HTML **address** element.

11 ToString

12

13 [C#] public const HtmlTextWriterTag Area;

14 [C++] public: const HtmlTextWriterTag Area;

15 [VB] Public Const Area As HtmlTextWriterTag

16 [JScript] public var Area : HtmlTextWriterTag;

17

18 *Description*

19 Specifies the HTML **area** element.

20 ToString

21

22 [C#] public const HtmlTextWriterTag B;

23 [C++] public: const HtmlTextWriterTag B;

24 [VB] Public Const B As HtmlTextWriterTag

25 [JScript] public var B : HtmlTextWriterTag;



*Description*

Specifies the HTML **b** element.

ToString

[C#]            public            const            HtmlTextWriterTag            Base;

[C++]            public:            const            HtmlTextWriterTag            Base;

[VB]            Public            Const            Base            As            HtmlTextWriterTag

[JScript]            public            var            Base            :            HtmlTextWriterTag;

*Description*

Specifies the HTML **base** element.

ToString

[C#]            public            const            HtmlTextWriterTag            Basefont;

[C++]            public:            const            HtmlTextWriterTag            Basefont;

[VB]            Public            Const            Basefont            As            HtmlTextWriterTag

[JScript]            public            var            Basefont            :            HtmlTextWriterTag;

*Description*

Specifies the HTML **basefont** element.

ToString

[C#]            public            const            HtmlTextWriterTag            Bdo;

[C++]            public:            const            HtmlTextWriterTag            Bdo;

1	[VB]	Public	Const	Bdo	As	HtmlTextWriterTag
2	[JScript]	public	var	Bdo	:	HtmlTextWriterTag;

3

4 *Description*

5       Specifies the HTML **bdo** element.

6       ToString

8	[C#]	public	const	HtmlTextWriterTag		Bgsound;
9	[C++]	public:	const	HtmlTextWriterTag		Bgsound;
10	[VB]	Public	Const	Bgsound	As	HtmlTextWriterTag
11	[JScript]	public	var	Bgsound	:	HtmlTextWriterTag;

12

13 *Description*

14       Specifies the HTML **bg sound** element.

15       ToString

17	[C#]	public	const	HtmlTextWriterTag		Big;
18	[C++]	public:	const	HtmlTextWriterTag		Big;
19	[VB]	Public	Const	Big	As	HtmlTextWriterTag
20	[JScript]	public	var	Big	:	HtmlTextWriterTag;

21

22 *Description*

23       Specifies the HTML **big** element.

24       ToString

25

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#]      public      const      HtmlTextWriterTag      Blockquote;  
[C++]     public:     const      HtmlTextWriterTag      Blockquote;  
[VB]      Public      Const      Blockquote      As      HtmlTextWriterTag  
[JScript] public      var      Blockquote      :      HtmlTextWriterTag;
```

*Description*  
Specifies the HTML **blockquote** element.  
ToString

```
[C#]      public      const      HtmlTextWriterTag      Body;  
[C++]     public:     const      HtmlTextWriterTag      Body;  
[VB]      Public      Const      Body      As      HtmlTextWriterTag  
[JScript] public      var      Body      :      HtmlTextWriterTag;
```

*Description*  
Specifies the HTML **body** element.  
ToString

```
[C#]      public      const      HtmlTextWriterTag      Br;  
[C++]     public:     const      HtmlTextWriterTag      Br;  
[VB]      Public      Const      Br      As      HtmlTextWriterTag  
[JScript] public      var      Br      :      HtmlTextWriterTag;
```

*Description*

1 Specifies the HTML **br** element.

2 ToString

3  
4 [C#] public const HtmlTextWriterTag Button;

5 [C++] public: const HtmlTextWriterTag Button;

6 [VB] Public Const Button As HtmlTextWriterTag

7 [JScript] public var Button : HtmlTextWriterTag;

8  
9 *Description*

10 Specifies the HTML **button** element.

11 ToString

12  
13 [C#] public const HtmlTextWriterTag Caption;

14 [C++] public: const HtmlTextWriterTag Caption;

15 [VB] Public Const Caption As HtmlTextWriterTag

16 [JScript] public var Caption : HtmlTextWriterTag;

17  
18 *Description*

19 Specifies the HTML **caption** element.

20 ToString

21  
22 [C#] public const HtmlTextWriterTag Center;

23 [C++] public: const HtmlTextWriterTag Center;

24 [VB] Public Const Center As HtmlTextWriterTag

25 [JScript] public var Center : HtmlTextWriterTag;

*Description*

Specifies the HTML **center** element.

ToString

[C#]            public            const            HtmlTextWriterTag            Cite;

[C++]           public:            const            HtmlTextWriterTag            Cite;

[VB]           Public            Const            Cite            As            HtmlTextWriterTag

[JScript]       public            var            Cite            :            HtmlTextWriterTag;

*Description*

Specifies the HTML **cite** element.

ToString

[C#]            public            const            HtmlTextWriterTag            Code;

[C++]           public:            const            HtmlTextWriterTag            Code;

[VB]           Public            Const            Code            As            HtmlTextWriterTag

[JScript]       public            var            Code            :            HtmlTextWriterTag;

*Description*

Specifies the HTML **code** element.

ToString

[C#]            public            const            HtmlTextWriterTag            Col;

[C++]           public:            const            HtmlTextWriterTag            Col;

1	[VB]	Public	Const	Col	As	HtmlTextWriterTag
2	[JScript]	public	var	Col	:	HtmlTextWriterTag;

3

4 *Description*

5       Specifies the HTML **col** element.

6       ToString

8	[C#]	public	const	HtmlTextWriterTag	Colgroup;	
9	[C++]	public:	const	HtmlTextWriterTag	Colgroup;	
10	[VB]	Public	Const	Colgroup	As	HtmlTextWriterTag
11	[JScript]	public	var	Colgroup	:	HtmlTextWriterTag;

12

13 *Description*

14       Specifies the HTML **colgroup** element.

15       ToString

17	[C#]	public	const	HtmlTextWriterTag	Dd;	
18	[C++]	public:	const	HtmlTextWriterTag	Dd;	
19	[VB]	Public	Const	Dd	As	HtmlTextWriterTag
20	[JScript]	public	var	Dd	:	HtmlTextWriterTag;

21

22 *Description*

23       Specifies the HTML **dd** element.

24       ToString

25

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#]      public      const      HtmlTextWriterTag      Del;  
[C++]     public:     const      HtmlTextWriterTag      Del;  
[VB]      Public      Const      Del      As      HtmlTextWriterTag  
[JScript] public      var      Del      :      HtmlTextWriterTag;
```

*Description*

Specifies the HTML **cel** element.  
ToString

```
[C#]      public      const      HtmlTextWriterTag      Dfn;  
[C++]     public:     const      HtmlTextWriterTag      Dfn;  
[VB]      Public      Const      Dfn      As      HtmlTextWriterTag  
[JScript] public      var      Dfn      :      HtmlTextWriterTag;
```

*Description*

Specifies the HTML **dfn** element.  
ToString

```
[C#]      public      const      HtmlTextWriterTag      Dir;  
[C++]     public:     const      HtmlTextWriterTag      Dir;  
[VB]      Public      Const      Dir      As      HtmlTextWriterTag  
[JScript] public      var      Dir      :      HtmlTextWriterTag;
```

*Description*

1 Specifies the HTML **dir** element.

2 ToString

3

4 [C#] public const HtmlTextWriterTag Div;

5 [C++] public: const HtmlTextWriterTag Div;

6 [VB] Public Const Div As HtmlTextWriterTag

7 [JScript] public var Div : HtmlTextWriterTag;

8

9 *Description*

10 Specifies the HTML **div** element.

11 ToString

12

13 [C#] public const HtmlTextWriterTag Dl;

14 [C++] public: const HtmlTextWriterTag Dl;

15 [VB] Public Const Dl As HtmlTextWriterTag

16 [JScript] public var Dl : HtmlTextWriterTag;

17

18 *Description*

19 Specifies the HTML **dl** element.

20 ToString

21

22 [C#] public const HtmlTextWriterTag Dt;

23 [C++] public: const HtmlTextWriterTag Dt;

24 [VB] Public Const Dt As HtmlTextWriterTag

25 [JScript] public var Dt : HtmlTextWriterTag;



*Description*

Specifies the HTML **dt** element.

ToString

[C#]            public            const            HtmlTextWriterTag            Em;

[C++]           public:            const            HtmlTextWriterTag            Em;

[VB]           Public            Const            Em            As            HtmlTextWriterTag

[JScript]       public            var            Em            :            HtmlTextWriterTag;

*Description*

Specifies the HTML **em** element.

ToString

[C#]            public            const            HtmlTextWriterTag            Embed;

[C++]           public:            const            HtmlTextWriterTag            Embed;

[VB]           Public            Const            Embed            As            HtmlTextWriterTag

[JScript]       public            var            Embed            :            HtmlTextWriterTag;

*Description*

Specifies the HTML **embed** element.

ToString

[C#]            public            const            HtmlTextWriterTag            Fieldset;

[C++]           public:            const            HtmlTextWriterTag            Fieldset;

1	[VB]	Public	Const	Fieldset	As	HtmlTextWriterTag
2	[JScript]	public	var	Fieldset	:	HtmlTextWriterTag;

3

4 *Description*

5       Specifies the HTML **fieldset** element.

6       ToString

8	[C#]	public	const	HtmlTextWriterTag	Font;	
9	[C++]	public:	const	HtmlTextWriterTag	Font;	
10	[VB]	Public	Const	Font	As	HtmlTextWriterTag
11	[JScript]	public	var	Font	:	HtmlTextWriterTag;

12

13 *Description*

14       Specifies the HTML **font** element.

15       ToString

17	[C#]	public	const	HtmlTextWriterTag	Form;	
18	[C++]	public:	const	HtmlTextWriterTag	Form;	
19	[VB]	Public	Const	Form	As	HtmlTextWriterTag
20	[JScript]	public	var	Form	:	HtmlTextWriterTag;

21

22 *Description*

23       Specifies the HTML **form** element.

24       ToString

25

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#]      public      const      HtmlTextWriterTag      Frame;  
[C++]     public:     const      HtmlTextWriterTag      Frame;  
[VB]      Public      Const      Frame      As      HtmlTextWriterTag  
[JScript] public      var      Frame      :      HtmlTextWriterTag;
```

*Description*

Specifies the HTML **frame** element.

ToString

```
[C#]      public      const      HtmlTextWriterTag      Frameset;  
[C++]     public:     const      HtmlTextWriterTag      Frameset;  
[VB]      Public      Const      Frameset      As      HtmlTextWriterTag  
[JScript] public      var      Frameset      :      HtmlTextWriterTag;
```

*Description*

Specifies the HTML **frameset** element.

ToString

```
[C#]      public      const      HtmlTextWriterTag      H1;  
[C++]     public:     const      HtmlTextWriterTag      H1;  
[VB]      Public      Const      H1      As      HtmlTextWriterTag  
[JScript] public      var      H1      :      HtmlTextWriterTag;
```

*Description*

1 Specifies the HTML **H1** element.

2 ToString

3

4 [C#] public const HtmlTextWriterTag H2;

5 [C++] public: const HtmlTextWriterTag H2;

6 [VB] Public Const H2 As HtmlTextWriterTag

7 [JScript] public var H2 : HtmlTextWriterTag;

8

9 *Description*

10 Specifies the HTML **H2** element.

11 ToString

12

13 [C#] public const HtmlTextWriterTag H3;

14 [C++] public: const HtmlTextWriterTag H3;

15 [VB] Public Const H3 As HtmlTextWriterTag

16 [JScript] public var H3 : HtmlTextWriterTag;

17

18 *Description*

19 Specifies the HTML **H3** element.

20 ToString

21

22 [C#] public const HtmlTextWriterTag H4;

23 [C++] public: const HtmlTextWriterTag H4;

24 [VB] Public Const H4 As HtmlTextWriterTag

25 [JScript] public var H4 : HtmlTextWriterTag;

*Description*

Specifies the HTML **H4** element.

ToString

[C#]            public            const            HtmlTextWriterTag            H5;

[C++]            public:            const            HtmlTextWriterTag            H5;

[VB]            Public            Const            H5            As            HtmlTextWriterTag

[JScript]            public            var            H5            :            HtmlTextWriterTag;

*Description*

Specifies the HTML **H5** element.

ToString

[C#]            public            const            HtmlTextWriterTag            H6;

[C++]            public:            const            HtmlTextWriterTag            H6;

[VB]            Public            Const            H6            As            HtmlTextWriterTag

[JScript]            public            var            H6            :            HtmlTextWriterTag;

*Description*

Specifies the HTML **H6** element.

ToString

[C#]            public            const            HtmlTextWriterTag            Head;

[C++]            public:            const            HtmlTextWriterTag            Head;

1	[VB]	Public	Const	Head	As	HtmlTextWriterTag
2	[JScript]	public	var	Head	:	HtmlTextWriterTag;

3

4 *Description*

5       Specifies the HTML **head** element.

6       ToString

8	[C#]	public	const		HtmlTextWriterTag	Hr;
9	[C++]	public:	const		HtmlTextWriterTag	Hr;
10	[VB]	Public	Const	Hr	As	HtmlTextWriterTag
11	[JScript]	public	var	Hr	:	HtmlTextWriterTag;

12

13 *Description*

14       Specifies the HTML **hr** element.

15       ToString

17	[C#]	public	const		HtmlTextWriterTag	Html;
18	[C++]	public:	const		HtmlTextWriterTag	Html;
19	[VB]	Public	Const	Html	As	HtmlTextWriterTag
20	[JScript]	public	var	Html	:	HtmlTextWriterTag;

21

22 *Description*

23       Specifies the HTML **html** element.

24       ToString

25

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#]      public      const      HtmlTextWriterTag      I;  
[C++]     public:     const      HtmlTextWriterTag      I;  
[VB]      Public      Const      I      As      HtmlTextWriterTag  
[JScript] public      var      I      :      HtmlTextWriterTag;
```

*Description*

Specifies the HTML **i** element.  
ToString

```
[C#]      public      const      HtmlTextWriterTag      Iframe;  
[C++]     public:     const      HtmlTextWriterTag      Iframe;  
[VB]      Public      Const      Iframe      As      HtmlTextWriterTag  
[JScript] public      var      Iframe      :      HtmlTextWriterTag;
```

*Description*

Specifies the HTML **iframe** element.  
ToString

```
[C#]      public      const      HtmlTextWriterTag      Img;  
[C++]     public:     const      HtmlTextWriterTag      Img;  
[VB]      Public      Const      Img      As      HtmlTextWriterTag  
[JScript] public      var      Img      :      HtmlTextWriterTag;
```

*Description*

1 Specifies the HTML **img** element.

2 ToString

3

4 [C#] public const HtmlTextWriterTag Input;

5 [C++] public: const HtmlTextWriterTag Input;

6 [VB] Public Const Input As HtmlTextWriterTag

7 [JScript] public var Input : HtmlTextWriterTag;

8

9 *Description*

10 Specifies the HTML **input** element.

11 ToString

12

13 [C#] public const HtmlTextWriterTag Ins;

14 [C++] public: const HtmlTextWriterTag Ins;

15 [VB] Public Const Ins As HtmlTextWriterTag

16 [JScript] public var Ins : HtmlTextWriterTag;

17

18 *Description*

19 Specifies the HTML **ins** element.

20 ToString

21

22 [C#] public const HtmlTextWriterTag Isindex;

23 [C++] public: const HtmlTextWriterTag Isindex;

24 [VB] Public Const Isindex As HtmlTextWriterTag

25 [JScript] public var Isindex : HtmlTextWriterTag;



*Description*

Specifies the HTML **isindex** element.

ToString

[C#]            public            const            HtmlTextWriterTag            Kbd;

[C++]           public:            const            HtmlTextWriterTag            Kbd;

[VB]           Public            Const            Kbd            As            HtmlTextWriterTag

[JScript]       public            var            Kbd            :            HtmlTextWriterTag;

*Description*

Specifies the HTML **kbd** element.

ToString

[C#]            public            const            HtmlTextWriterTag            Label;

[C++]           public:            const            HtmlTextWriterTag            Label;

[VB]           Public            Const            Label            As            HtmlTextWriterTag

[JScript]       public            var            Label            :            HtmlTextWriterTag;

*Description*

Specifies the HTML **label** element.

ToString

[C#]            public            const            HtmlTextWriterTag            Legend;

[C++]           public:            const            HtmlTextWriterTag            Legend;

1 [VB] Public Const Legend As HtmlTextWriterTag  
 2 [JScript] public var Legend : HtmlTextWriterTag;

3  
 4 *Description*

5 Specifies the HTML **legend** element.

6 ToString

7  
 8 [C#] public const HtmlTextWriterTag Li;

9 [C++] public: const HtmlTextWriterTag Li;

10 [VB] Public Const Li As HtmlTextWriterTag

11 [JScript] public var Li : HtmlTextWriterTag;

12  
 13 *Description*

14 Specifies the HTML **li** element.

15 ToString

16  
 17 [C#] public const HtmlTextWriterTag Link;

18 [C++] public: const HtmlTextWriterTag Link;

19 [VB] Public Const Link As HtmlTextWriterTag

20 [JScript] public var Link : HtmlTextWriterTag;

21  
 22 *Description*

23 Specifies the HTML **link** element.

24 ToString

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#]      public      const      HtmlTextWriterTag      Map;  
[C++]     public:     const      HtmlTextWriterTag      Map;  
[VB]      Public      Const      Map      As      HtmlTextWriterTag  
[JScript] public      var      Map      :      HtmlTextWriterTag;
```

*Description*  
Specifies the HTML **map** element.  
ToString

```
[C#]      public      const      HtmlTextWriterTag      Marquee;  
[C++]     public:     const      HtmlTextWriterTag      Marquee;  
[VB]      Public      Const      Marquee      As      HtmlTextWriterTag  
[JScript] public      var      Marquee      :      HtmlTextWriterTag;
```

*Description*  
Specifies the HTML **marquee** element.  
ToString

```
[C#]      public      const      HtmlTextWriterTag      Menu;  
[C++]     public:     const      HtmlTextWriterTag      Menu;  
[VB]      Public      Const      Menu      As      HtmlTextWriterTag  
[JScript] public      var      Menu      :      HtmlTextWriterTag;
```

*Description*

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25

2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25

5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25

7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

16  
17  
18  
19  
20  
21  
22  
23  
24  
25

18  
19  
20  
21  
22  
23  
24  
25

19  
20  
21  
22  
23  
24  
25

20  
21  
22  
23  
24  
25

22  
23  
24  
25

23  
24  
25

24

25

25

*Description*

Specifies the HTML **noframes** element.

ToString

[C#]            public            const            HtmlTextWriterTag            Noscript;

[C++]           public:            const            HtmlTextWriterTag            Noscript;

[VB]           Public            Const            Noscript            As            HtmlTextWriterTag

[JScript]       public            var            Noscript            :            HtmlTextWriterTag;

*Description*

Specifies the HTML **noscript** element.

ToString

[C#]            public            const            HtmlTextWriterTag            Object;

[C++]           public:            const            HtmlTextWriterTag            Object;

[VB]           Public            Const            Object            As            HtmlTextWriterTag

[JScript]       public            var            Object            :            HtmlTextWriterTag;

*Description*

Specifies the HTML **object** element.

ToString

[C#]            public            const            HtmlTextWriterTag            Ol;

[C++]           public:            const            HtmlTextWriterTag            Ol;

1	[VB]	Public	Const	Ol	As	HtmlTextWriterTag
2	[JScript]	public	var	Ol	:	HtmlTextWriterTag;

3

4 *Description*

5       Specifies the HTML **ol** element.

6       ToString

8	[C#]	public	const	HtmlTextWriterTag	Option;
9	[C++]	public:	const	HtmlTextWriterTag	Option;

10	[VB]	Public	Const	Option	As	HtmlTextWriterTag
11	[JScript]	public	var	Option	:	HtmlTextWriterTag;

12

13 *Description*

14       Specifies the HTML **option** element.

15       ToString

17	[C#]	public	const	HtmlTextWriterTag	P;
18	[C++]	public:	const	HtmlTextWriterTag	P;

19	[VB]	Public	Const	P	As	HtmlTextWriterTag
20	[JScript]	public	var	P	:	HtmlTextWriterTag;

21

22 *Description*

23       Specifies the HTML **p** element.

24       ToString

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#]      public      const      HtmlTextWriterTag      Param;  
[C++]     public:     const      HtmlTextWriterTag      Param;  
[VB]      Public      Const      Param      As      HtmlTextWriterTag  
[JScript] public      var      Param      :      HtmlTextWriterTag;
```

*Description*

Specifies the HTML **param** element.  
ToString

```
[C#]      public      const      HtmlTextWriterTag      Pre;  
[C++]     public:     const      HtmlTextWriterTag      Pre;  
[VB]      Public      Const      Pre      As      HtmlTextWriterTag  
[JScript] public      var      Pre      :      HtmlTextWriterTag;
```

*Description*

Specifies the HTML **pre** element.  
ToString

```
[C#]      public      const      HtmlTextWriterTag      Q;  
[C++]     public:     const      HtmlTextWriterTag      Q;  
[VB]      Public      Const      Q      As      HtmlTextWriterTag  
[JScript] public      var      Q      :      HtmlTextWriterTag;
```

*Description*

1 Specifies the HTML **q** element.

2 ToString

3  
4 [C#] public const HtmlTextWriterTag Rt;

5 [C++] public: const HtmlTextWriterTag Rt;

6 [VB] Public Const Rt As HtmlTextWriterTag

7 [JScript] public var Rt : HtmlTextWriterTag;

8  
9 *Description*

10 Specifies the HTML **rt** element.

11 ToString

12  
13 [C#] public const HtmlTextWriterTag Ruby;

14 [C++] public: const HtmlTextWriterTag Ruby;

15 [VB] Public Const Ruby As HtmlTextWriterTag

16 [JScript] public var Ruby : HtmlTextWriterTag;

17  
18 *Description*

19 Specifies the HTML **ruby** element.

20 ToString

21  
22 [C#] public const HtmlTextWriterTag S;

23 [C++] public: const HtmlTextWriterTag S;

24 [VB] Public Const S As HtmlTextWriterTag

25 [JScript] public var S : HtmlTextWriterTag;



1  
2 *Description*

3 Specifies the HTML **s** element.

4 ToString

5  
6 [C#] public const HtmlTextWriterTag Samp;

7 [C++] public: const HtmlTextWriterTag Samp;

8 [VB] Public Const Samp As HtmlTextWriterTag

9 [JScript] public var Samp : HtmlTextWriterTag;

10  
11 *Description*

12 Specifies the HTML **samp** element.

13 ToString

14  
15 [C#] public const HtmlTextWriterTag Script;

16 [C++] public: const HtmlTextWriterTag Script;

17 [VB] Public Const Script As HtmlTextWriterTag

18 [JScript] public var Script : HtmlTextWriterTag;

19  
20 *Description*

21 Specifies the HTML **script** element.

22 ToString

23  
24 [C#] public const HtmlTextWriterTag Select;

25 [C++] public: const HtmlTextWriterTag Select;

1	[VB]	Public	Const	Select	As	HtmlTextWriterTag
2	[JScript]	public	var	Select	:	HtmlTextWriterTag;

3

4 *Description*

5       Specifies the HTML **select** element.

6       ToString

7

8	[C#]	public	const	HtmlTextWriterTag	Small;
---	------	--------	-------	-------------------	--------

9	[C++]	public:	const	HtmlTextWriterTag	Small;
---	-------	---------	-------	-------------------	--------

10	[VB]	Public	Const	Small	As	HtmlTextWriterTag
----	------	--------	-------	-------	----	-------------------

11	[JScript]	public	var	Small	:	HtmlTextWriterTag;
----	-----------	--------	-----	-------	---	--------------------

12

13 *Description*

14       Specifies the HTML **small** element.

15       ToString

16

17	[C#]	public	const	HtmlTextWriterTag	Span;
----	------	--------	-------	-------------------	-------

18	[C++]	public:	const	HtmlTextWriterTag	Span;
----	-------	---------	-------	-------------------	-------

19	[VB]	Public	Const	Span	As	HtmlTextWriterTag
----	------	--------	-------	------	----	-------------------

20	[JScript]	public	var	Span	:	HtmlTextWriterTag;
----	-----------	--------	-----	------	---	--------------------

21

22 *Description*

23       Specifies the HTML **span** element.

24       ToString

25

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#]      public      const      HtmlTextWriterTag      Strike;  
[C++]     public:     const      HtmlTextWriterTag      Strike;  
[VB]      Public      Const      Strike      As      HtmlTextWriterTag  
[JScript] public      var      Strike      :      HtmlTextWriterTag;
```

*Description*

Specifies the HTML **strike** element.

ToString

```
[C#]      public      const      HtmlTextWriterTag      Strong;  
[C++]     public:     const      HtmlTextWriterTag      Strong;  
[VB]      Public      Const      Strong      As      HtmlTextWriterTag  
[JScript] public      var      Strong      :      HtmlTextWriterTag;
```

*Description*

Specifies the HTML **strong** element.

ToString

```
[C#]      public      const      HtmlTextWriterTag      Style;  
[C++]     public:     const      HtmlTextWriterTag      Style;  
[VB]      Public      Const      Style      As      HtmlTextWriterTag  
[JScript] public      var      Style      :      HtmlTextWriterTag;
```

*Description*

1 Specifies the HTML **style** element.

2 ToString

3  
4 [C#] public const HtmlTextWriterTag Sub;

5 [C++] public: const HtmlTextWriterTag Sub;

6 [VB] Public Const Sub As HtmlTextWriterTag

7 [JScript] public var Sub : HtmlTextWriterTag;

8  
9 *Description*

10 Specifies the HTML **sub** element.

11 ToString

12  
13 [C#] public const HtmlTextWriterTag Sup;

14 [C++] public: const HtmlTextWriterTag Sup;

15 [VB] Public Const Sup As HtmlTextWriterTag

16 [JScript] public var Sup : HtmlTextWriterTag;

17  
18 *Description*

19 Specifies the HTML **sup** element.

20 ToString

21  
22 [C#] public const HtmlTextWriterTag Table;

23 [C++] public: const HtmlTextWriterTag Table;

24 [VB] Public Const Table As HtmlTextWriterTag

25 [JScript] public var Table : HtmlTextWriterTag;

1  
2 *Description*

3 Specifies the HTML **table** element.

4 ToString

5  
6 [C#] public const HtmlTextWriterTag Tbody;

7 [C++] public: const HtmlTextWriterTag Tbody;

8 [VB] Public Const Tbody As HtmlTextWriterTag

9 [JScript] public var Tbody : HtmlTextWriterTag;

10  
11 *Description*

12 Specifies the HTML **tbody** element.

13 ToString

14  
15 [C#] public const HtmlTextWriterTag Td;

16 [C++] public: const HtmlTextWriterTag Td;

17 [VB] Public Const Td As HtmlTextWriterTag

18 [JScript] public var Td : HtmlTextWriterTag;

19  
20 *Description*

21 Specifies the HTML **td** element.

22 ToString

23  
24 [C#] public const HtmlTextWriterTag Textarea;

25 [C++] public: const HtmlTextWriterTag Textarea;

1	[VB]	Public	Const	Textarea	As	HtmlTextWriterTag
2	[JScript]	public	var	Textarea	:	HtmlTextWriterTag;

3

4 *Description*

5       Specifies the HTML **textarea** element.

6       ToString

8	[C#]	public	const	HtmlTextWriterTag	Tfoot;	
9	[C++]	public:	const	HtmlTextWriterTag	Tfoot;	
10	[VB]	Public	Const	Tfoot	As	HtmlTextWriterTag
11	[JScript]	public	var	Tfoot	:	HtmlTextWriterTag;

12

13 *Description*

14       Specifies the HTML **tfoot** element.

15       ToString

17	[C#]	public	const	HtmlTextWriterTag	Th;	
18	[C++]	public:	const	HtmlTextWriterTag	Th;	
19	[VB]	Public	Const	Th	As	HtmlTextWriterTag
20	[JScript]	public	var	Th	:	HtmlTextWriterTag;

21

22 *Description*

23       Specifies the HTML **th** element.

24       ToString

25

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#]      public      const      HtmlTextWriterTag      Thead;  
[C++]     public:     const      HtmlTextWriterTag      Thead;  
[VB]      Public      Const      Thead      As      HtmlTextWriterTag  
[JScript] public      var      Thead      :      HtmlTextWriterTag;
```

*Description*  
Specifies the HTML **thead** element.  
ToString

```
[C#]      public      const      HtmlTextWriterTag      Title;  
[C++]     public:     const      HtmlTextWriterTag      Title;  
[VB]      Public      Const      Title      As      HtmlTextWriterTag  
[JScript] public      var      Title      :      HtmlTextWriterTag;
```

*Description*  
Specifies the HTML **title** element.  
ToString

```
[C#]      public      const      HtmlTextWriterTag      Tr;  
[C++]     public:     const      HtmlTextWriterTag      Tr;  
[VB]      Public      Const      Tr      As      HtmlTextWriterTag  
[JScript] public      var      Tr      :      HtmlTextWriterTag;
```

*Description*

1 Specifies the HTML **tr** element.

2 ToString

3  
4 [C#] public const HtmlTextWriterTag Tt;

5 [C++] public: const HtmlTextWriterTag Tt;

6 [VB] Public Const Tt As HtmlTextWriterTag

7 [JScript] public var Tt : HtmlTextWriterTag;

8  
9 *Description*

10 Specifies the HTML **tt** element.

11 ToString

12  
13 [C#] public const HtmlTextWriterTag U;

14 [C++] public: const HtmlTextWriterTag U;

15 [VB] Public Const U As HtmlTextWriterTag

16 [JScript] public var U : HtmlTextWriterTag;

17  
18 *Description*

19 Specifies the HTML **u** element.

20 ToString

21  
22 [C#] public const HtmlTextWriterTag Ul;

23 [C++] public: const HtmlTextWriterTag Ul;

24 [VB] Public Const Ul As HtmlTextWriterTag

25 [JScript] public var Ul : HtmlTextWriterTag;



*Description*

Specifies the HTML **ul** element.

ToString

[C#]            public            const            HtmlTextWriterTag            Unknown;

[C++]           public:            const            HtmlTextWriterTag            Unknown;

[VB]           Public           Const           Unknown           As           HtmlTextWriterTag

[JScript]       public           var           Unknown           :           HtmlTextWriterTag;

*Description*

The **System.String** passed as an HTML tag is not recognized.

ToString

[C#]            public            const            HtmlTextWriterTag            Var;

[C++]           public:            const            HtmlTextWriterTag            Var;

[VB]           Public           Const           Var           As           HtmlTextWriterTag

[JScript]       public           var           Var           :           HtmlTextWriterTag;

*Description*

Specifies the HTML **var** element.

ToString

[C#]            public            const            HtmlTextWriterTag            Wbr;

[C++]           public:            const            HtmlTextWriterTag            Wbr;

1	[VB]	Public	Const	Wbr	As	HtmlTextWriterTag
2	[JScript]	public	var	Wbr	:	HtmlTextWriterTag;

3

4 *Description*

5 Specifies the HTML **wbr** element.

6 ToString

8	[C#]	public	const	HtmlTextWriterTag	Xml;
9	[C++]	public:	const	HtmlTextWriterTag	Xml;
10	[VB]	Public	Const	Xml	As HtmlTextWriterTag
11	[JScript]	public	var	Xml	: HtmlTextWriterTag;

12

13 *Description*

14 Specifies the HTML **xml** element.

15 IAttributeAccessor interface (System.Web.UI)

16 ToString

17

18

19 *Description*

20 Defines methods used by ASP.NET server controls to provide

21 programmatic access to any attribute declared in the opening tag of a server

22 control.

23 If you author a custom server control that inherits from the

24 **System.Web.UI.WebControls.WebControl** ,

25 **System.Web.UI.HtmlControls.HtmlControl** , or

**System.Web.UI.WebControls.ListItem** class, the .NET Framework automatically provides programmatic access to attributes since each of these classes implement the **IAttributeAccessor** interface.

#### GetAttribute

```
[C#]          string          GetAttribute(string          key);
[C++]         String*         GetAttribute(String*         key);
[VB]  Function GetAttribute(ByVal key As String) As String
[JScript]     function  GetAttribute(key : String) : String;
```

#### *Description*

When implemented by a class, retrieves the specified attribute property from the server control.

*Return Value:* The value of the specified attribute. A **System.String** object that represents the name of the server control attribute.

#### SetAttribute

```
[C#]          void          SetAttribute(string          key,          string          value);
[C++]         void          SetAttribute(String*         key,          String*         value);
[VB]  Sub      SetAttribute(ByVal key As String, ByVal value As String)
[JScript]     function  SetAttribute(key : String, value : String);
```

#### *Description*

When implemented by a class, designates an attribute and its value to assign to the ASP.NET server control. The name of the attribute to be set. The value assigned to the attribute.

IDataBindingsAccessor interface (System.Web.UI)

SetAttribute

#### *Description*

Allows access to the collection of databinding expressions on a control at design time.

DataBindings

SetAttribute

[C#]                    DataBindingCollection                    DataBindings                    {get;}

[C++]                    DataBindingCollection\*                    get\_DataBindings();

[VB]    ReadOnly    Property    DataBindings    As    DataBindingCollection

[JScript]    abstract    function    get    DataBindings()    :    DataBindingCollection;

#### *Description*

Indicates a collection of all data bindings on the control. This property is read-only.

HasDataBindings

SetAttribute

[C#]                    bool                    HasDataBindings                    {get;}

1 [C++] bool get\_HasDataBindings();  
 2 [VB] ReadOnly Property HasDataBindings As Boolean  
 3 [JScript] abstract function get HasDataBindings() : Boolean;

#### 5 *Description*

6 Returns whether the control contains any data binding logic. This method is  
 7 only accessed by RAD designers.

8 *Return Value:* **true** if the control contains data binding logic; otherwise, **false** .

9 ImageClickEventArgs class (System.Web.UI)

10 SetAttribute

#### 13 *Description*

14 Provides data for any events that occur when a user clicks on a image-based  
 15 ASP.NET server control, such as the  
 16 **System.Web.UI.HtmlControls.HtmlInputImage** or  
 17 **System.Web.UI.WebControls.Image** server controls. This class cannot be  
 18 inherited.

19 This class passes the location where a user clicked on an **HtmlInputImage**  
 20 server control or an **Image** Web server control. Clicking an **HtmlInputImage**  
 21 server control causes a **ServerClick** event to fire, while clicking an **Image** server  
 22 control causes a **Click** event to occur. You can then use event handlers to  
 23 programmatically respond to the event based on the value of these coordinates.

24 SetAttribute

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#]          public          int          X;
[C++]          public:          int          X;
[VB]          Public          X          As          Integer
[JScript]      public          var          X          :          int;
```

*Description*

An integer that represents the x-coordinate where a user clicked on an image-based ASP.NET server control.

SetAttribute

```
[C#]          public          int          Y;
[C++]          public:          int          Y;
[VB]          Public          Y          As          Integer
[JScript]      public          var          Y          :          int;
```

*Description*

An integer that represents the y-coordinate where a user clicked on an image-based ASP.NET server control.

ImageClickEventArgs

*Example Syntax:*

SetAttribute

```
[C#]          public          ImageClickEventArgs(int          x,          int          y);
[C++]          public:          ImageClickEventArgs(int          x,          int          y);
```

1 [VB] Public Sub New(ByVal x As Integer, ByVal y As Integer)  
2 [JScript] public function ImageClickEventArgs(x : int, y : int);

3  
4 *Description*

5 Initializes a new instance of the **System.Web.UI.ImageClickEventArgs**  
6 class. The x-coordinate where the user clicked on an image-based ASP.NET  
7 server control. The y-coordinate where the user clicked on an image-based  
8 ASP.NET server control.

9 ImageClickEventHandler delegate (System.Web.UI)

10 ToString  
11  
12

13 *Description*

14 Represents the method that will handle any events that are raised when a  
15 user clicks on an image-based ASP.NET server control. The server control that is  
16 the source of the event. An **System.Web.UI.ImageClickEventArgs** object that  
17 contains event data.

18 This delegate defines the required signature for all click events raised when  
19 a user clicks on an image-based ASP.NET server control. You must use the  
20 signature defined by this delegate for any **OnServerClick** event handlers that you  
21 create for an **System.Web.UI.HtmlControls.HtmlInputImage** server control or  
22 any **OnClick** event handlers that you create for an  
23 **System.Web.UI.WebControls.ImageButton** Web server control.

24 INamingContainer interface (System.Web.UI)

25 ToString

1  
2  
3 *Description*

4 Identifies a container control that creates a new ID namespace within a  
5 **System.Web.UI.Page** object's control hierarchy. This is a marker interface only.

6 Any control that implements this interface creates a new namespace in  
7 which all child control ID attributes are guaranteed to be unique within an entire  
8 application. The marker provided by this interface allows unique naming of the  
9 dynamically generated server control instances within the Web server controls that  
10 support data binding. These controls include the  
11 **System.Web.UI.WebControls.Repeater** control, the  
12 **System.Web.UI.WebControls.DataGrid** control, or the  
13 **System.Web.UI.WebControls.RadioButtonList** control.

14 **IParserAccessor** interface (System.Web.UI)

15 **ToString**  
16  
17

18 *Description*

19 Defines the method that ASP.NET server controls must implement to  
20 recognize when elements, either HTML or XML, are parsed.

21 When elements are parsed, they are recognized as children of the server  
22 control that implements this interface. As such, they are added to the control's  
23 **System.Web.UI.Control.Controls** property. These elements, generally speaking,  
24 are added as **System.Web.UI.LiteralControl** objects to the  
25 **System.Web.UI.ControlCollection** object associated with **Controls** property.



## AddParsedSubObject

```
[C#]          void          AddParsedSubObject(object          obj);
[C++]          void          AddParsedSubObject(Object*          obj);
[VB]    Sub    AddParsedSubObject(ByVal    obj    As    Object)
[JScript]      function      AddParsedSubObject(obj      :      Object);
```

### *Description*

When implemented by an ASP.NET server control, notifies the server control that an element, either XML or HTML, was parsed, and adds the element to the server control's **System.Web.UI.ControlCollection** object. The **System.Object** instance to add to the parent control.

IPostBackDataHandler interface (System.Web.UI)

## AddParsedSubObject

### *Description*

Defines methods that ASP.NET server controls must implement to automatically load post back data.

If you want a server control you design to examine form data that is posted back to the server by the client, you must implement the **IPostBackDataHandler** interface. The contract that this interface defines allows a server control to determine whether its state should be altered as a result of the post back, and to raise the appropriate events.

## LoadPostData

```

1
2 [C#]    bool    LoadPostData(string    postDataKey,    NameValueCollection
3 postCollection);
4 [C++]    bool    LoadPostData(String*    postDataKey,    NameValueCollection*
5 postCollection);
6 [VB]    Function    LoadPostData(ByVal    postDataKey    As    String,    ByVal
7 postCollection    As    NameValueCollection)    As    Boolean
8 [JScript]    function    LoadPostData(postDataKey    :    String,    postCollection    :
9 NameValueCollection)    :    Boolean;

```

#### *Description*

When implemented by a class, processes post back data for an ASP.NET server control.

**Return Value:** **true** if the server control's state changes as a result of the post back; otherwise **false**.

The Web Forms page framework tracks all of the server control's that return true to this method call, then invokes the **System.Web.UI.IPostBackDataHandler.RaisePostDataChangedEvent** on those controls. The key identifier for the control. The collection of all incoming name values.

#### **RaisePostDataChangedEvent**

```

23 [C#]                void                RaisePostDataChangedEvent();
24 [C++]                void                RaisePostDataChangedEvent();
25 [VB]                Sub                RaisePostDataChangedEvent()

```

1 [JScript] function RaisePostDataChangedEvent();

2  
3 *Description*

4 When implemented by a class, signals the server control object to notify the  
5 ASP.NET application that the state of the control has changed.

6 Change events for the server control that implements this interface, if any,  
7 are raised from this method.

8 IPostBackEventHandler interface (System.Web.UI)

9 RaisePostDataChangedEvent

10  
11  
12 *Description*

13 Defines the method ASP.NET server controls must implement to handle  
14 post back events.

15 To create a server control that captures form submit information from the  
16 browser, you must implement this interface. For more information on how to use  
17 this interface, see .

18 RaisePostBackEvent

19  
20 [C#] void RaisePostBackEvent(string eventArgument);

21 [C++] void RaisePostBackEvent(String\* eventArgument);

22 [VB] Sub RaisePostBackEvent(ByVal eventArgument As String)

23 [JScript] function RaisePostBackEvent(eventArgument : String);

24  
25 *Description*

When implemented by a class, enables a server control to process an event raised when a form is posted to the server.

This method provides the functionality for many events implemented by HTML and Web server controls. A **System.String** that represents an optional event argument to be passed to the event handler.

StateManager interface (System.Web.UI)

RaisePostBackEvent

### *Description*

Defines the properties and methods any class must implement to support view state management for a server control.

A server control's view state comprises the cumulative values of the control's properties. This interface includes methods that save and load a server control's view state values, as well as a method that instructs the control to track any changes to its view state.

IsTrackingViewState

RaisePostBackEvent

[C#]                    bool                    IsTrackingViewState                    {get;}

[C++]                    bool                    get\_IsTrackingViewState();

[VB]    ReadOnly    Property    IsTrackingViewState    As    Boolean

[JScript]    abstract    function    get    IsTrackingViewState()    :    Boolean;

### *Description*

When implemented by a class, gets a value indicating whether a server control is tracking its view state changes.

### LoadViewState

```
[C#]          void          LoadViewState(object          state);
[C++]          void          LoadViewState(Object*          state);
[VB]      Sub      LoadViewState(ByVal      state      As      Object)
[JScript]      function      LoadViewState(state      :      Object);
```

#### *Description*

When implemented by a class, loads the server control's previously saved view state to the control. An **System.Object** that contains the saved view state values for the control.

### SaveViewState

```
[C#]          object          SaveViewState();
[C++]          Object*          SaveViewState();
[VB]      Function      SaveViewState()      As      Object
[JScript]      function      SaveViewState()      :      Object;
```

#### *Description*

When implemented by a class, saves the changes to a server control's view state to an **System.Object**.

*Return Value:* The **Object** that contains the view state changes.

### TrackViewState

[C#]	void	TrackViewState();
[C++]	void	TrackViewState();
[VB]	Sub	TrackViewState()
[JScript]	function	TrackViewState();

#### *Description*

When implemented by a class, instructs the server control to track changes to its view state.

Once this method has been called on a server control, the **System.Web.UI.IStateManager.IsTrackingViewState** property will return **true**

ITagNameToTypeMapper interface (System.Web.UI)

TrackViewState

#### *Description*

Maps a sequence of text characters to a .NET Framework type when an .aspx file is processed on the server.

Classes that implement this interface strongly type any string in an .aspx file to the appropriate class. HTML and literal text strings are mapped to the type **System.Web.UI.LiteralControl** , HTML anchor tags with a **runat="server"** attribute/value pair are mapped to the type **System.Web.UI.HtmlControls.HtmlAnchor** , while tags are mapped to the type **System.Web.UI.WebControls.Button** .

## GetControlType

```
[C#]    Type    GetControlType(string    tagName,    IDictionary    attrs);  
[C++]   Type*   GetControlType(String*   tagName,   IDictionary*   attrs);  
[VB]    Function GetControlType(ByVal tagName As String, ByVal attrs As  
IDictionary)                                         As                                         Type  
[JScript] function GetControlType(tagName : String, attrs : IDictionary) : Type;
```

### *Description*

Retrieves the .NET Framework type that processes the control declared in the .aspx file.

*Return Value:* The .NET Framework type that is assigned to the control. The element name of the control sent from the .aspx file. A collection of the attributes on the control in the .aspx file.

ITemplate interface (System.Web.UI)

## GetControlType

### *Description*

Defines the method to implement for populating an ASP.NET server control with child controls when using a the control with inline templates when declared in an .aspx file.

## InstantiateIn

```
[C#]    void    InstantiateIn(Control    container);
```

```

1 [C++]          void          InstantiateIn(Control*          container);
2 [VB]      Sub      InstantiateIn(ByVal      container      As      Control)
3 [JScript]      function      InstantiateIn(container      :      Control);
4

```

#### *Description*

When implemented by a class, defines the **System.Web.UI.Control** object that child controls and templates belong to. These child controls are in turn defined within an inline template.

When developing templated server controls you do not need to implement this method, the .NET Framework provides the implementation for you. The **Control** object to contain the instantiated controls from the inline template.

Validator interface (System.Web.UI)

InstantiateIn

#### *Description*

Defines the properties and methods that objects that participate in validation must implement.

Classes that implement this interface represent a possible user input error. When the **System.Web.UI.Validator.Validate** method is called, the class updates its **System.Web.UI.Validator.IsValid** property to signify whether the error occurred. The **System.Web.UI.Validator.ErrorMessage** property contains a text description of the error condition that you can display when the error occurs.

ErrorMessage

InstantiateIn



```

1
2 [C#]          string          ErrorMessage          {get;          set;}
3 [C++]      String*      get_ErrorMessage();void      set_ErrorMessage(String*);
4 [VB]          Property          ErrorMessage          As          String
5 [JScript] abstract function get ErrorMessage() : String;public abstract function set
6 ErrorMessage(String);
7

```

#### 8 *Description*

9 When implemented by a class, gets or sets the error message text generated  
10 when the condition being validated fails.

11 Web Forms page developers access this property declaratively in the  
12 opening tag of a validation server control. For more information, see .

13 IsValid

14 InstantiateIn

```

15
16 [C#]          bool          IsValid          {get;          set;}
17 [C++]      bool          get_IsValid();void          set_IsValid(bool);
18 [VB]          Property          IsValid          As          Boolean
19 [JScript] abstract function get IsValid() : Boolean;public abstract function set
20 IsValid(Boolean);
21

```

#### 22 *Description*

23 When implemented by a class, gets or sets a value indicating whether the  
24 user-entered content in the specified control passes validation.

25 Validate



```

1 [C++]                public:                LiteralControl();
2 [VB]                  Public                  Sub                  New()
3 [JScript] public function LiteralControl(); Initializes a new instance of the
4 System.Web.UI.LiteralControl class.

```

#### *Description*

Initializes a new instance of the **System.Web.UI.LiteralControl** class that contains a literal string to be rendered on the requested ASP.NET page.

LiteralControl

#### *Example Syntax:*

Validate

```

13 [C#]                public                LiteralControl(string                text);
14 [C++]                public:                LiteralControl(String*                text);
15 [VB]                Public                Sub                New(ByVal                text                As                String)
16 [JScript]                public                function                LiteralControl(text                :                String);

```

#### *Description*

Initializes a new instance of the **System.Web.UI.LiteralControl** class with the specified text. The text to be rendered on the requested Web page.

ChildControlsCreated

ClientID

Context

Controls

EnableViewState

100120 00000000

- 1 Events
- 2 HasChildViewState
- 3 ID
- 4 IsTrackingViewState
- 5 NamingContainer
- 6 Page
- 7 Parent
- 8 Site
- 9 TemplateSourceDirectory
- 10 Text
- 11 Validate

14 *Description*

15 Gets or sets the text content of the **System.Web.UI.LiteralControl** object.

16 UniqueID

17 ViewState

18 ViewStateIgnoresCase

19 Visible

20 CreateControlCollection

```
22 [C#]    protected    override    ControlCollection    CreateControlCollection();
23 [C++]    protected:    ControlCollection*    CreateControlCollection();
24 [VB]    Overrides    Protected    Function    CreateControlCollection()    As
25 ControlCollection
```

1 [JScript] protected override function CreateControlCollection() :  
2 ControlCollection;

3  
4 *Description*

5 Creates an **System.Web.UI.EmptyControlCollection** object for the  
6 current instance of the **System.Web.UI.LiteralControl**.

7 *Return Value:* The **EmptyControlCollection** object for the current control.

8 By default, **LiteralControl** objects contain only text and no child server  
9 controls. If you want to change this behavior, you must override this method.

10 **Render**

11  
12 [C#] protected override void Render(HtmlTextWriter output);

13 [C++] protected: void Render(HtmlTextWriter\* output);

14 [VB] Overrides Protected Sub Render(ByVal output As HtmlTextWriter)

15 [JScript] protected override function Render(output : HtmlTextWriter);  
16

17 *Description*

18 Writes the content of the **System.Web.UI.LiteralControl** object to the  
19 ASP.NET page.

20 When you create a custom server control and want to render specific  
21 HTML or text to a client, you can improve the performance of the control by  
22 passing the value of the **System.Web.UI.LiteralControl.Text** property to the  
23 **System.Web.UI.Control.Render(System.Web.UI.HtmlTextWriter)** method  
24 rather than calling the **LiteralControl.Render** method. An  
25



```

1
2 [C#]      public      object      Deserialize(Stream      stream);
3 [C++]      public:      Object*      Deserialize(Stream*      stream);
4 [VB] Public Function Deserialize(ByVal stream As Stream) As Object
5 [JScript] public function Deserialize(stream : Stream) : Object; Deserializes an
6 LOS-formatted object.

```

#### 8 *Description*

9 Deserializes a LOS-formatted object from a **System.IO.Stream** object.

10 *Return Value:* Returns the deserialized object. The source of the object to be

11 deserialized.

#### 12 *Deserialize*

```

13
14 [C#]      public      object      Deserialize(string      input);
15 [C++]      public:      Object*      Deserialize(String*      input);
16 [VB] Public Function Deserialize(ByVal input As String) As Object
17 [JScript] public function Deserialize(input : String) : Object;
18

```

#### 19 *Description*

20 Deserializes a LOS formatted object from a string.

21 *Return Value:* Returns the deserialized object. The source of the object to be

22 deserialized.

#### 23 *Deserialize*

```

24
25 [C#]      public      object      Deserialize(TextReader      input);

```

```

1 [C++]      public:      Object*      Deserialize(TextReader*      input);
2 [VB] Public Function Deserialize(ByVal input As TextReader) As Object
3 [JScript] public function Deserialize(input : TextReader) : Object;

```

#### *Description*

Deserializes a LOS-formatted object from a **System.IO.TextReader** object.

*Return Value:* Returns the deserialized object. The source of the object to be deserialized.

#### Serialize

```

12 [C#]      public      void      Serialize(Stream      stream,      object      value);
13 [C++]      public:      void      Serialize(Stream*      stream,      Object*      value);
14 [VB] Public Sub Serialize(ByVal stream As Stream, ByVal value As Object)
15 [JScript] public function Serialize(stream : Stream, value : Object); Serializes the
16 Web          Forms          view          state          value.

```

#### *Description*

Serializes the Web Forms view state value into a **System.IO.Stream** object. The object to serialize into. The view state information to pass.

#### Serialize

```

23 [C#]      public      void      Serialize(TextWriter      output,      object      value);
24 [C++]      public:      void      Serialize(TextWriter*      output,      Object*      value);
25 [VB] Public Sub Serialize(ByVal output As TextWriter, ByVal value As Object)

```



1 [JScript] public function Serialize(output : TextWriter, value : Object);

2  
3 *Description*

4 Serializes the view state value into a **System.IO.TextWriter** object. The  
5 object to serialize into. The view state information to pass.

6 ObjectConverter class (System.Web.UI)

7 ToString

8  
9  
10 *Description*

11 ObjectConverter

12 *Example Syntax:*

13 ToString

14  
15 [C#] public ObjectConverter();

16 [C++] public: ObjectConverter();

17 [VB] Public Sub New()

18 [JScript] public function ObjectConverter();

19 ConvertValue

20  
21 [C#] public static object ConvertValue(object value, Type toType, string  
22 formatString);

23 [C++] public: static Object\* ConvertValue(Object\* value, Type\* toType, String\*  
24 formatString);

25 [VB] Public Shared Function ConvertValue(ByVal value As Object, ByVal

```

1 toType As Type, ByVal formatString As String) As Object
2 [JScript] public static function ConvertValue(value : Object, toType : Type,
3 formatString : String) : Object;

```

4  
5 *Description*

6 ObjectTagBuilder class (System.Web.UI)

7 ToString

8  
9  
10 *Description*

11 ObjectTagBuilder

12 *Example Syntax:*

13 ToString

14  
15 [C#] public ObjectTagBuilder();

16 [C++] public: ObjectTagBuilder();

17 [VB] Public Sub New()

18 [JScript] public function ObjectTagBuilder();

19 ControlType

20 FChildrenAsProperties

21 FIsNonParserAccessor

22 HasAspCode

23 ID

24 InDesigner

25 NamingContainerType

1       Parser  
2       TagName  
3       AppendLiteralString  
4  
5 [C#]     public     override     void     AppendLiteralString(string     s);  
6 [C++]     public:     void     AppendLiteralString(String\*     s);  
7 [VB]   Overrides   Public   Sub   AppendLiteralString(ByVal s As String)  
8 [JScript]   public   override   function   AppendLiteralString(s : String);  
9

10   *Description*

11       AppendSubBuilder  
12  
13 [C#]   public   override   void   AppendSubBuilder(ControlBuilder subBuilder);  
14 [C++]   public:   void   AppendSubBuilder(ControlBuilder\* subBuilder);  
15 [VB]   Overrides   Public   Sub   AppendSubBuilder(ByVal subBuilder As  
16 ControlBuilder)  
17 [JScript]   public   override   function   AppendSubBuilder(subBuilder :  
18 ControlBuilder);  
19

20   *Description*

21       Init  
22  
23 [C#]   public   override   void   Init(TemplateParser parser, ControlBuilder  
24 parentBuilder, Type type, string tagName, string id, IDictionary attribs);  
25 [C++]   public:   void   Init(TemplateParser\* parser, ControlBuilder\* parentBuilder,

```

1 Type* type, String* tagName, String* id, IDictionary* attrs);
2 [VB] Overrides Public Sub Init(ByVal parser As TemplateParser, ByVal
3 parentBuilder As ControlBuilder, ByVal type As Type, ByVal tagName As String,
4 ByVal id As String, ByVal attrs As IDictionary)
5 [JScript] public override function Init(parser : TemplateParser, parentBuilder :
6 ControlBuilder, type : Type, tagName : String, id : String, attrs : IDictionary);

```

#### Description

OutputCacheLocation enumeration (System.Web.UI)

ToString

#### Description

Specifies the valid values for the location of the output cache.

The values specified by this enumeration are used when you include an @OutputCache directive in a .aspx file. These values set the cacheability of page output. For more information, see .

ToString

[C#]	public	const	OutputCacheLocation	Any;
[C++]	public:	const	OutputCacheLocation	Any;
[VB]	Public	Const	Any As	OutputCacheLocation
[JScript]	public	var	Any :	OutputCacheLocation;

#### Description

1 The output cache can be located on the browser client where the request  
2 originated, on a proxy server (or any other server) participating in the request, or  
3 on the server where the request was processed.

4 ToString

6 [C#] public const OutputCacheLocation Client;

7 [C++] public: const OutputCacheLocation Client;

8 [VB] Public Const Client As OutputCacheLocation

9 [JScript] public var Client : OutputCacheLocation;

11 *Description*

12 The output cache is located on the browser client where the request  
13 originated.

14 ToString

16 [C#] public const OutputCacheLocation Downstream;

17 [C++] public: const OutputCacheLocation Downstream;

18 [VB] Public Const Downstream As OutputCacheLocation

19 [JScript] public var Downstream : OutputCacheLocation;

21 *Description*

22 The output cache can be stored in any HTTP 1.1 cache-capable devices  
23 other than the origin server. This include proxy servers and the clietn that made  
24 the request.

25 ToString

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

[C#]	public	const	OutputCacheLocation	None;
[C++]	public:	const	OutputCacheLocation	None;
[VB]	Public	Const	None	As OutputCacheLocation
[JScript]	public	var	None	: OutputCacheLocation;

*Description*

The output cache is disabled for the requested page.

ToString

[C#]	public	const	OutputCacheLocation	Server;
[C++]	public:	const	OutputCacheLocation	Server;
[VB]	Public	Const	Server	As OutputCacheLocation
[JScript]	public	var	Server	: OutputCacheLocation;

*Description*

The output cache is located on the Web server where the request was processed.

Page class (System.Web.UI)

ToString

*Description*

Represents an .aspx file requested from a server that hosts an ASP.NET Web application.

The **Page** class is associated with files that have an .aspx extension. These files are compiled at runtime as a **Page** object and cached in server memory.

### ToString

[C#]	protected	const	string	postEventArgumentID;
[C++]	protected:	const	String*	postEventArgumentID;
[VB]	Protected	Const	postEventArgumentID	As String
[JScript]	protected	var	postEventArgumentID	: String;

### Description

#### ToString

[C#]	protected	const	string	postEventSourceID;
[C++]	protected:	const	String*	postEventSourceID;
[VB]	Protected	Const	postEventSourceID	As String
[JScript]	protected	var	postEventSourceID	: String;

### Description

#### Page

#### Example Syntax:

#### ToString

[C#]	public	Page();
[C++]	public:	Page();
[VB]	Public	Sub New()

1 [JScript] public function Page();

2  
3 *Description*

4 Initializes a new instance of the **System.Web.UI.Page** class.

5 The default constructor initializes all fields to their default values.

6 Application

7 ToString

8  
9 [C#] public HttpApplicationState Application {get;}

10 [C++] public: \_\_property HttpApplicationState\* get\_Application();

11 [VB] Public ReadOnly Property Application As HttpApplicationState

12 [JScript] public function get Application() : HttpApplicationState;

13  
14 *Description*

15 Gets the **Application** object for the current Web request.

16 AspCompatMode

17 ToString

18  
19 [C#] bool AspCompatMode {set;}

20 [C++] protected: \_\_property void set\_AspCompatMode(bool);

21 [VB] Property AspCompatMode As Boolean

22 [JScript] protected function set AspCompatMode(Boolean);

23  
24 *Description*



1       Sets a value indicating whether the page can be executed on a single-  
2 threaded apartment (STA) thread.

3       When set to **true** , this allows the page to be executed on a single-threaded  
4 apartment (STA) thread. This allows the page to call STA components, such as  
5 components developed with Visual Basic 6.0. Setting this to **true** also allows the  
6 page to call COM+ 1.0 components that require access to the unmanaged ASP  
7 built-in objects. These are accessible through the ASP **ObjectContext** object or  
8 the **OnStartPage** method.

9       AutoHandlers

10      Buffer

11      ToString

12  
13  
14      *Description*

15       Sets a value indicating whether the page output is buffered.

16       In most circumstances, do not set this property in code. Set the **buffer**  
17 attribute to **true** using the directive in the .aspx file. When the page is requested,  
18 the dynamically generated class sets the property.

19      Cache

20      ToString

21  
22      [C#]           public           Cache           Cache           {get;}  
23      [C++]           public:           \_\_property       Cache\*           get\_Cache();  
24      [VB]           Public       ReadOnly       Property       Cache       As       Cache  
25      [JScript]       public       function       get       Cache()       :       Cache;

1  
2 *Description*

3 Gets the **System.Web.Caching.Cache** object associated with the  
4 application in which the page resides.

5 An application's **Cache** object allows you to store and retrieve arbitrary  
6 data on subsequent requests . The cache is not specifically associated with a page  
7 or user session. It is used primarily to enhance application performance. For more  
8 information, see .

9 ChildControlsCreated

10 ClientID

11 ClientTarget

12 ToString

13  
14  
15 *Description*

16 Gets or sets a value that allows you to override automatic detection of  
17 browser capabilities and to specify how a page renders for particular browser  
18 clients.

19 If you do not set this property, the **System.Web.HttpBrowserCapabilities**  
20 object associated with the **System.Web.UI.Page.Request** property reflects the  
21 capabilities of the client browser.

22 CodePage

23 ToString

24  
25 [C#] int CodePage {set;}

1	[C++]	protected:	__property	void	set_CodePage(int);
2	[VB]	Property	CodePage	As	Integer
3	[JScript]	protected	function	set	CodePage(int);

4

5 *Description*

6       Sets the code page identifier for the current **System.Web.UI.Page** .

7       In most circumstances, do not set this property in code. Set the **CodePage**  
8 attribute to the value you want using the directive in the .aspx file. When the page  
9 is requested, the dynamically generated class sets the property.

10       ContentType

11       ToString

12					
13	[C#]	string	ContentType	{set;}	
14	[C++]	protected:	__property	void	set_ContentType(String*);
15	[VB]	Property	ContentType	As	String
16	[JScript]	protected	function	set	ContentType(String);

17

18 *Description*

19       Sets the HTTP MIME type for the **System.Web.HttpResponse** object  
20 associated with the page.

21       In most circumstances, do not set this property in code. Set the  
22 **ContentType** attribute using the directive in the .aspx file. When the page is  
23 requested, the dynamically generated class sets the property.

24       Context

25       ToString

```

1
2 [C#]      protected      override      HttpContext      Context      {get;}
3 [C++]      protected:      __property      virtual      HttpContext*      get_Context();
4 [VB]      Overrides      Protected      ReadOnly      Property      Context      As      HttpContext
5 [JScript]      protected      function      get      Context()      :      HttpContext;
6

```

### *Description*

Gets the **System.Web.HttpContext** object associated with the page.

This property provides programmatic access to the context the page runs in, including information about the request, response, session and application.

Controls

Culture

ToString

### *Description*

Sets the culture ID for the **System.Threading.Thread** object associated with the page.

In most circumstances, do not set this property in code. Set the **Culture** attribute in the directive in the .aspx file. When the page is requested, the dynamically generated class sets the property.

EnableViewState

ToString

```

25 [C#]      public      override      bool      EnableViewState      {get;      set;}

```

```

1 [C++] public: __property virtual bool get_EnableViewState();public: __property
2 virtual void set_EnableViewState(bool);
3 [VB] Overrides Public Property EnableViewState As Boolean
4 [JScript] public function get EnableViewState() : Boolean;public function set
5 EnableViewState(Boolean);

```

### *Description*

Gets or sets a value indicating whether the page maintains its view state, and the view state of any server controls it contains, when the current page request ends.

EnableViewStateMac

ToString

```

14 [C#] protected bool EnableViewStateMac {get; set;}
15 [C++] protected: __property bool get_EnableViewStateMac();protected:
16 __property void set_EnableViewStateMac(bool);
17 [VB] Protected Property EnableViewStateMac As Boolean
18 [JScript] protected function get EnableViewStateMac() : Boolean;protected
19 function set EnableViewStateMac(Boolean);

```

### *Description*

Gets or sets a value indicating whether ASP.NET should run a machine authentication check (MAC) on the page's view state when the page is posted back from the client.

1 In most circumstances, do not set this property in code. Set the  
2 **EnableViewStateMac** attribute to **true** using the directive in the .aspx file. When  
3 the page is requested, the dynamically generated class sets the property.

4       ErrorPage

5       ToString

6  
7 [C#]       public       string       ErrorPage       {get;       set;}

8 [C++] public: \_\_property String\* get\_ErrorPage();public: \_\_property void  
9 set\_ErrorPage(String\*);

10 [VB]       Public       Property       ErrorPage       As       String

11 [JScript] public function get ErrorPage() : String;public function set  
12 ErrorPage(String);

13  
14 *Description*

15 Gets or sets the error page to which the requesting browser should be  
16 redirected in the event of an unhandled page exception.

17       Events

18       FileDependencies

19       ToString

20  
21  
22 *Description*

23 Sets an array of files that the current **System.Web.HttpResponse** object is  
24 dependent upon.

1 In most circumstances, do not set this property in code. Set the  
2 **FileDependencies** attribute to **true** using the directive in the .aspx file. When the  
3 page is requested, the dynamically generated class sets the property.

4 HasChildViewState

5 ID

6 ToString

7  
8  
9 *Description*

10 Gets or sets an identifier for a particular instance of the  
11 **System.Web.UI.Page** class.

12 IsPostBack

13 ToString

14  
15 [C#] public bool IsPostBack {get;}

16 [C++] public: \_\_property bool get\_IsPostBack();

17 [VB] Public ReadOnly Property IsPostBack As Boolean

18 [JScript] public function get IsPostBack() : Boolean;

19  
20 *Description*

21 Gets a value indicating whether the page is being loaded in response to a  
22 client postback, or if it is being loaded and accessed for the first time.

23 IsReusable

24 ToString

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#]          public          bool          IsReusable          {get;}
[C++]          public:          __property          bool          get_IsReusable();
[VB]    Public    ReadOnly    Property    IsReusable    As    Boolean
[JScript]    public    function    get    IsReusable()    :    Boolean;
```

*Description*

Page class can be cached/reused  
IsTrackingViewState  
IsValid  
ToString

*Description*

Gets a value indicating whether page validation succeeded.  
For this property to return **true** , all validation server controls in the  
**System.Web.UI.Page.Validators** property must validate successfully.

LCID  
ToString

```
[C#]          int          LCID          {set;}
[C++]          protected:          __property          void          set_LCID(int);
[VB]          Property          LCID          As          Integer
[JScript]          protected          function          set          LCID(int);
```



## Description

Sets the locale identifier for the **System.Threading.Thread** object associated with the page.

In most circumstances, do not set this property in code. Set the **LCID** attribute in the directive in the .aspx file. When the page is requested, the dynamically generated class sets the property.

NamingContainer

Page

Parent

Request

ToString

## Description

Gets the **System.Web.HttpRequest** object for the requested page.

The **Request** object contains information about current incoming HTTP request.

Response

ToString

```
[C#]      public      HttpResponse      Response      {get;}
[C++]     public:      __property      HttpResponse*      get_Response();
[VB]      Public      ReadOnly      Property      Response      As      HttpResponse
[JScript] public      function      get      Response()      :      HttpResponse;
```

## Description

Gets the **System.Web.HttpResponse** object associated with the **System.Web.UI.Page**. This object allows you to send HTTP response data to a client, and contains information about that response.

ResponseEncoding

ToString

[C#]                    string                    ResponseEncoding                    {set;}

[C++]    protected:    \_\_property    void    set\_ResponseEncoding(String\*);

[VB]            Property            ResponseEncoding            As            String

[JScript]    protected    function    set    ResponseEncoding(String);

## Description

Sets the encoding language for the current **System.Web.HttpResponse** object.

In most circumstances, do not set this property in code. Set the **ResponseEncoding** attribute to the value you want using the directive in the .aspx file. When the page is requested, the dynamically generated class sets the property.

Server

ToString

[C#]            public            HttpServerUtility            Server            {get;}

[C++]    public:            \_\_property            HttpServerUtility\*            get\_Server();

[VB]    Public    ReadOnly    Property    Server    As    HttpServerUtility

1 [JScript] public function get Server() : HttpServerUtility;

2  
3 *Description*

4 Gets the **Server** object, which is an instance of the  
5 **System.Web.HttpServerUtility** class.

6 This property provides access to the frequently used  
7 **System.Web.HttpServerUtility.HtmlEncode(System.String)** and  
8 **System.Web.HttpServerUtility.MapPath(System.String)** methods, among  
9 others.

10 Session

11 ToString

12  
13 [C#] public virtual HttpSessionState Session {get;}

14 [C++] public: \_\_property virtual HttpSessionState\* get\_Session();

15 [VB] Overridable Public ReadOnly Property Session As HttpSessionState

16 [JScript] public function get Session() : HttpSessionState;

17  
18 *Description*

19 Gets the current **Session** object provided by ASP.NET.

20 This property provides information about the current request's session. A  
21 **Session** object is maintained for each user that requests a page or document from  
22 an ASP.NET application. Variables stored in the **Session** object are not discarded  
23 when the user moves from page to page in the application; instead, these variables  
24 persist as long as the user is accessing pages in your application. For more  
25 information about session state, see .

1 Site  
2 SmartNavigation  
3 ToString  
4  
5

6 *Description*

7 Gets or sets a value indicating whether smart navigation is enabled.  
8 In most circumstances, do not set this property in code. Set the  
9 **SmartNavigation** attribute to **true** in the directive in the .aspx file. When the page  
10 is requested, the dynamically generated class sets this property.

11 SupportAutoEvents  
12 TemplateSourceDirectory  
13 Trace  
14 ToString  
15  
16

17 *Description*

18 Gets the **System.Web.TraceContext** object for the current Web request.  
19 Tracing tracks and presents the execution details about a Web request. For  
20 trace data to be visible in a rendered page, you must enable tracing at the page or  
21 application level.

22 TraceEnabled  
23 ToString  
24

25 [C#] bool TraceEnabled {set;}

1	[C++]	protected:	__property	void	set_TraceEnabled(bool);
2	[VB]	Property	TraceEnabled	As	Boolean
3	[JScript]	protected	function	set	TraceEnabled(Boolean);

4

5 *Description*

6       Sets a value indicating whether tracing is enabled for the

7 **System.Web.UI.Page** .

8       In most circumstances, do not set this property in code. Set the **Trace**

9 attribute to **true** in the directive in the .aspx file. When the page is requested, the

10 dynamically generated class sets the property.

11       TraceModeValue

12       ToString

13

14	[C#]	TraceMode	TraceModeValue	{set;}
----	------	-----------	----------------	--------

15	[C++]	protected:	__property	void	set_TraceModeValue(TraceMode);
16	[VB]	Property	TraceModeValue	As	TraceMode
17	[JScript]	protected	function	set	TraceModeValue(TraceMode);

18

19 *Description*

20       Sets the mode in which trace statements are displayed on the page.

21       In most circumstances, do not set this property in code. Set the **TraceMode**

22 attribute in the directive in the .aspx file. When the page is requested, the

23 dynamically generated class sets the property.

24       TransactionMode

25       ToString

1						
2	[C#]	int	TransactionMode	{set;}		
3	[C++]	protected: __property	void	set_TransactionMode(int);		
4	[VB]	Property	TransactionMode	As Integer		
5	[JScript]	protected	function	set TransactionMode(int);		

6

7 *Description*

8       Sets the level of transaction support for the page.

9       In most circumstances, do not set this property in code. Set the

10 **Transaction** attribute in the directive in the .aspx file. When the page is requested,

11 the dynamically generated class sets the property.

12       UICulture

13       ToString

14						
15	[C#]	string	UICulture	{set;}		
16	[C++]	protected: __property	void	set_UICulture(String*);		
17	[VB]	Property	UICulture	As String		
18	[JScript]	protected	function	set UICulture(String);		

19

20 *Description*

21       Sets the UI ID for the **System.Threading.Thread** object associated with

22 the page.

23       This property is a shortcut for **System.Threading.Thread.CurrentThread**

24 . The culture is a property of the executing thread

25       UniqueID

1 User  
2 ToString

3  
4  
5 *Description*

6 Gets information about the user making the page request.

7 An **IPrincipal** object represents the security context of the user on whose  
8 behalf the code is running, including that user's identity and any roles to which  
9 they belong.

10 Validators

11 ToString

12  
13 [C#] public ValidatorCollection Validators {get;}

14 [C++] public: \_\_property ValidatorCollection\* get\_Validators();

15 [VB] Public ReadOnly Property Validators As ValidatorCollection

16 [JScript] public function get Validators() : ValidatorCollection;

17  
18 *Description*

19 Gets a collection of all validation controls contained on the requested page.

20 ViewState

21 ViewStateIgnoresCase

22 Visible

23 ToString

24

25

1  
2  
3 *Description*

4 Gets or sets a value indicating whether the **System.Web.UI.Page** object  
5 should be rendered.

6 **AspCompatBeginProcessRequest**

7  
8 [C#] protected IAsyncResult AspCompatBeginProcessRequest(HttpContext  
9 context, AsyncCallback cb, object extraData);

10 [C++] protected: IAsyncResult\* AspCompatBeginProcessRequest(HttpContext\*  
11 context, AsyncCallback\* cb, Object\* extraData);

12 [VB] Protected Function AspCompatBeginProcessRequest(ByVal context As  
13 HttpContext, ByVal cb As AsyncCallback, ByVal extraData As Object) As  
14 IAsyncResult

15 [JScript] protected function AspCompatBeginProcessRequest(context :  
16 HttpContext, cb : AsyncCallback, extraData : Object) : IAsyncResult;

17  
18 *Description*

19 Initiates a request for Active Server Page (ASP) resources. This method is  
20 provided for compatibility with legacy ASP applications.

21 Do not call this method. An **System.Web.HttpContext** object with  
22 information about the current request. The callback method. Any extra data  
23 needed to process the request in the same manner as an ASP request.

24 **AspCompatEndProcessRequest**  
25



```

1
2 [C#] protected void AspCompatEndProcessRequest(IAsyncResult result);
3 [C++] protected: void AspCompatEndProcessRequest(IAsyncResult* result);
4 [VB] Protected Sub AspCompatEndProcessRequest(ByVal result As
5 IAsyncResult)
6 [JScript] protected function AspCompatEndProcessRequest(result :
7 IAsyncResult);
8

```

### *Description*

Terminates a request for Active Server Page (ASP) resources. This method is provided for compatibility with legacy ASP applications.

Do not call this method. The ASP page generated by the request.

CreateHtmlTextWriter

```

15 [C#] protected virtual HtmlTextWriter CreateHtmlTextWriter(TextWriter tw);
16 [C++] protected: virtual HtmlTextWriter* CreateHtmlTextWriter(TextWriter*
17 tw);
18 [VB] Overridable Protected Function CreateHtmlTextWriter(ByVal tw As
19 TextWriter) As HtmlTextWriter
20 [JScript] protected function CreateHtmlTextWriter(tw : TextWriter) :
21 HtmlTextWriter;
22

```

### *Description*

Creates an **System.Web.UI.HtmlTextWriter** object to render the page's content.

*Return Value:* The specified **System.Web.UI.HtmlTextWriter** or **System.Web.UI.Html32TextWriter** object.

The object created is based on information from the **System.Web.HttpBrowserCapabilities** object associated with the page request. Usually, this object is an **System.Web.UI.HtmlTextWriter** or **System.Web.UI.Html32TextWriter** object. For derived pages, you can override this method to create a custom text writer. Creates an **System.Web.UI.HtmlTextWriter** object to render the page's content. If the **IsUplevel** property is set to **false**, an **Html32TextWriter** object is created to render requests originating from downlevel browsers. For derived pages, you can override this method to create a custom text writer. The text writer to create.

#### DesignerInitialize

[C#]	public	void	DesignerInitialize();
[C++]	public:	void	DesignerInitialize();
[VB]	Public	Sub	DesignerInitialize()
[JScript]	public	function	DesignerInitialize();

#### Description

Performs any initialization of the instance of the **System.Web.UI.Page** class that is required by RAD designers. This method is only used at design time.

#### DeterminePostBackMode

[C#]	protected	virtual	NameValueCollection	DeterminePostBackMode();
[C++]	protected:	virtual	NameValueCollection*	DeterminePostBackMode();

```

1 [VB] Overridable Protected Function DeterminePostBackMode() As
2 NameValueCollection
3 [JScript] protected function DeterminePostBackMode() : NameValueCollection;
4

```

### *Description*

Determines the type of request made for the **Page** class. This information is based on whether the page was posted back, and whether the GET or POST HTTP method was used for the request.

*Return Value:* If the postback used the POST method, the Form information is returned from the **Context** object. If the postback used the GET method, the query string information is returned. If the page is being requested for the first time, **null** is returned.

### GetPostBackClientEvent

```

15 [C#] public string GetPostBackClientEvent(Control control, string argument);
16 [C++] public: String* GetPostBackClientEvent(Control* control, String*
17 argument);
18 [VB] Public Function GetPostBackClientEvent(ByVal control As Control, ByVal
19 argument As String) As String
20 [JScript] public function GetPostBackClientEvent(control : Control, argument :
21 String) : String;
22

```

### *Description*

Obtains a reference to a client-side script function that causes, when invoked, a server postback to the form.

*Return Value:* The **String** that represents the client event. The server control that receives the client event postback. A **System.String** argument that is passed to the **System.Web.UI.IPostBackEventHandler.RaisePostBackEvent(System.String)** method.

#### GetPostBackClientHyperlink

[C#] public string GetPostBackClientHyperlink(Control control, string argument);

[C++] public: String\* GetPostBackClientHyperlink(Control\* control, String\* argument);

[VB] Public Function GetPostBackClientHyperlink(ByVal control As Control, ByVal argument As String) As String

[JScript] public function GetPostBackClientHyperlink(control : Control, argument : String) : String;

#### Description

Appends **javascript:** to the beginning of the return from a **System.Web.UI.Page.GetPostBackEventReference(System.Web.UI.Control)** call to allow hyperlink post back processing on the server.

*Return Value:* The name of the client-side function and the ID of the server control that processed the function and argument passed to the control. The server control to process the postback. The parameter passed to the server control.

#### GetPostBackEventReference

[C#] public string GetPostBackEventReference(Control control);

[C++] public: String\* GetPostBackEventReference(Control\* control);

1 [VB] Public Function GetPostBackEventReference(ByVal control As Control) As  
2 String

3 [JScript] public function GetPostBackEventReference(control : Control) : String;

4 Obtains a reference to a client-side script function that causes, when invoked, the  
5 server to postback to the page.

6  
7 *Description*

8 Obtains a reference to a client-side script function that causes, when  
9 invoked, the server to postback to the page.

10 *Return Value:* The text of the client-side function call that can be inserted into a  
11 client-side event handler. The server control to process the postback on the server.

12 GetPostBackEventReference

13  
14 [C#] public string GetPostBackEventReference(Control control, string argument);

15 [C++] public: String\* GetPostBackEventReference(Control\* control, String\*  
16 argument);

17 [VB] Public Function GetPostBackEventReference(ByVal control As Control,  
18 ByVal argument As String) As String

19 [JScript] public function GetPostBackEventReference(control : Control, argument  
20 : String) : String;

21  
22 *Description*

23 Obtains a reference to a client-side script function that causes, when  
24 invoked, the server to postback to the page. It also passes a parameter to the server  
25 control that performs the postback processing on the server.

*Return Value:* The text of the client-side function call that can be inserted into a client-side event handler. The server control to process the postback. The parameter passed to the server control.

#### GetTypeHashCode

```
[C#]          public          virtual          int          GetTypeHashCode();
[C++]          public:          virtual          int          GetTypeHashCode();
[VB]  Overridable  Public  Function  GetTypeHashCode()  As  Integer
[JScript]      public      function      GetTypeHashCode()      :      int;
```

#### Description

Retrieves a hash code that is generated by **Page** objects that are generated at run time. This hash code is unique to the **Page** object's control hierarchy.

*Return Value:* The hash code generated at run time. The default is 0.

#### InitOutputCache

```
[C#] protected virtual void InitOutputCache(int duration, string varyByHeader,
string varyByCustom, OutputCacheLocation location, string varyByParam);
[C++] protected: virtual void InitOutputCache(int duration, String*
varyByHeader, String* varyByCustom, OutputCacheLocation location, String*
varyByParam);
[VB] Overridable Protected Sub InitOutputCache(ByVal duration As Integer,
ByVal varyByHeader As String, ByVal varyByCustom As String, ByVal location
As OutputCacheLocation, ByVal varyByParam As String)
[JScript] protected function InitOutputCache(duration : int, varyByHeader : String,
```

1 varyByCustom : String, location : OutputCacheLocation, varyByParam : String);

2  
3 *Description*

4       Initializes the output cache for the current page request.

5       You should not call this method. To enable and manipulate output caching  
6 for a page, use either the directive in the .aspx file, or the methods and properties  
7 of the **System.Web.HttpCachePolicy** class. The latter are accessible through  
8 **Response.Cache** syntax in the page's code-declaration block or code-behind file.  
9 For more information, see . An integer representing the amount of time that  
10 objects stored in the output cache are valid. A semi-colon separated list of headers  
11 that content from the output cache will vary by. A string that represents the **Vary**  
12 HTTP header. The location of the output cache as specified by the  
13 **System.Web.UI.OutputCacheLocation** enumeration. A semi-colon separated list  
14 of parameters, received by a GET or POST method, that content from the output  
15 cache will vary by.

16       IsClientScriptBlockRegistered

17  
18 [C#]       public       bool       IsClientScriptBlockRegistered(string       key);

19 [C++]       public:       bool       IsClientScriptBlockRegistered(String\*       key);

20 [VB] Public Function IsClientScriptBlockRegistered(ByVal key As String) As  
21 Boolean

22 [JScript] public function IsClientScriptBlockRegistered(key : String) : Boolean;

23  
24 *Description*

Determines if the client script block is registered with the page.  
*Return Value:* Returns **true** if the script block is registered; otherwise, **false**.

Call this method before calling **System.Web.UI.Page.RegisterClientScriptBlock(System.String,System.String)** to avoid unnecessarily assembling the client-side script. This is particularly important if the script requires a large amount of server resources to create. The string key of the client script to search for.

#### IsStartupScriptRegistered

[C#] public bool IsStartupScriptRegistered(string key);

[C++] public: bool IsStartupScriptRegistered(String\* key);

[VB] Public Function IsStartupScriptRegistered(ByVal key As String) As Boolean

[JScript] public function IsStartupScriptRegistered(key : String) : Boolean;

#### Description

Determines if the client startup script is registered with the **System.Web.UI.Page** object.

*Return Value:* **true** if the startup script is registered; otherwise, **false**.

Call this method before calling **System.Web.UI.Page.RegisterStartupScript(System.String,System.String)** to avoid unnecessarily assembling the client-side script. This is particularly important if the script requires a large amount of server resources to create. The string key of the startup script to search for.

#### LoadPageStateFromPersistenceMedium



```

1
2 [C#]    protected    virtual    object    LoadPageStateFromPersistenceMedium();
3 [C++]    protected:    virtual    Object*    LoadPageStateFromPersistenceMedium();
4 [VB]    Overridable    Protected    Function    LoadPageStateFromPersistenceMedium()
5 As                                             Object
6 [JScript]    protected    function    LoadPageStateFromPersistenceMedium() : Object;
7

```

### 8 *Description*

9 Loads any saved view-state information to the **System.Web.UI.Page**  
10 object. Override this method if you want to load the **Page** view state in anything  
11 other than a hidden field.

12 *Return Value:* The saved view state.

13 If you want to specify something other than hidden fields to save view state  
14 when using this method, you must also override the  
15 **SaveStateToPersistenceMedium** method.

### 16 **MapPath**

```

17
18 [C#]    public    string    MapPath(string    virtualPath);
19 [C++]    public:    String*    MapPath(String*    virtualPath);
20 [VB]    Public    Function    MapPath(ByVal    virtualPath    As    String)    As    String
21 [JScript]    public    function    MapPath(virtualPath    :    String)    :    String;
22

```

### 23 *Description*

24 Retrieves the physical path that a virtual path, either absolute or relative,  
25 maps to.

1 *Return Value:* The physical path associated with the virtual path. A **System.String**  
2 that represents the virtual path.

### 3       ProcessRequest

4  
5 [C#]       public       void       ProcessRequest(HttpContext       context);  
6 [C++]   public:   \_\_sealed   void   ProcessRequest(HttpContext\*   context);  
7 [VB] NotOverridable Public Sub ProcessRequest(ByVal context As HttpContext)  
8 [JScript]   public   function   ProcessRequest(context   :   HttpContext);  
9

### 10      *Description*

#### 11       RaisePostBackEvent

12  
13 [C#]   protected   virtual   void   RaisePostBackEvent(IPostBackEventHandler  
14 sourceControl,                               string                               eventArgument);  
15 [C++]   protected:   virtual   void   RaisePostBackEvent(IPostBackEventHandler\*  
16 sourceControl,                               String\*                               eventArgument);  
17 [VB] Overridable Protected Sub RaisePostBackEvent(ByVal sourceControl As  
18 IPostBackEventHandler,   ByVal   eventArgument   As   String)  
19 [JScript]   protected   function   RaisePostBackEvent(sourceControl   :  
20 IPostBackEventHandler,                       eventArgument                       :           String);  
21

### 22      *Description*

23       Notifies the server control that caused postback that it should handle an  
24 incoming postback event.  
25

1 The **Page** calls this when a post back occurs. This occurs in the page  
2 lifecycle after loading and change notification have completed, but before  
3 prerendering occurs. The ASP.NET server control that caused postback. This  
4 control must implement the **IPostBackEventHandler** interface. The post back  
5 argument.

#### 6 RegisterArrayDeclaration

8 [C#] public void RegisterArrayDeclaration(string arrayName, string arrayValue);

9 [C++] public: void RegisterArrayDeclaration(String\* arrayName, String\*  
10 arrayValue);

11 [VB] Public Sub RegisterArrayDeclaration(ByVal arrayName As String, ByVal  
12 arrayValue As String)

13 [JScript] public function RegisterArrayDeclaration(arrayName : String,  
14 arrayValue : String);

#### 16 *Description*

17 Declares a value that will be declared as an ECMAScript array declaration  
18 when the page renders. This can be used by script-based controls to declare  
19 themselves within an array so that a client script library can work with all the  
20 controls of the same type. The name of the array in which to declare the value. The  
21 value to place in the array.

#### 22 RegisterClientScriptBlock

24 [C#] public virtual void RegisterClientScriptBlock(string key, string script);

25 [C++] public: virtual void RegisterClientScriptBlock(String\* key, String\* script);

```

1 [VB] Overridable Public Sub RegisterClientScriptBlock(ByVal key As String,
2   ByVal          script          As          String)
3 [JScript] public function RegisterClientScriptBlock(key : String, script : String);
4

```

#### *Description*

Allows ASP.NET server controls to emit client-side script blocks into the **System.Web.UI.Page** .

The client-side script is emitted jut after the opening tag of the **Page** object's

Top of Form  
 element. The script block is emitted as you define to the output stream, so you must include both tags of the

Bottom of Form

### **System.Web.UI.Design**

#### *Description*

The **System.Web.UI.Design** namespace contains classes that can be used to extend design-time support for Web Forms.

CalendarDataBindingHandler class (System.Web.UI.Design)

#### *Description*

Provides a data binding handler for calendar data.

Constructors:

CalendarDataBindingHandler

*Example Syntax:*

[C#] public CalendarDataBindingHandler();

[C++] public: CalendarDataBindingHandler();

[VB] Public Sub New()

[JScript] public function CalendarDataBindingHandler();

Methods:

DataBindControl

[C#] public override void DataBindControl(IDesignerHost designerHost, Control control);

[C++] public: void DataBindControl(IDesignerHost\* designerHost, Control\* control);

[VB] Overrides Public Sub DataBindControl(ByVal designerHost As IDesignerHost, ByVal control As Control)

[JScript] public override function DataBindControl(designerHost : IDesignerHost, control : Control);

### *Description*

Adds this data binding to the specified control. The designer host for the document that contains the control. The control to add this data binding to.

ColorBuilder class (System.Web.UI.Design)

ToString

*Description*

Launches a color editor that allows a user to select a color.

The

**System.Web.UI.Design.ColorBuilder.BuildColor(System.ComponentModel.IComponent,System.Windows.Forms.Control,System.String)** method launches a user interface for selecting a color value.

BuildColor

[C#] public static string BuildColor(IComponent component, Control owner, string initialColor);

[C++] public: static String\* BuildColor(IComponent\* component, Control\* owner, String\* initialColor);

[VB] Public Shared Function BuildColor(ByVal component As IComponent, ByVal owner As Control, ByVal initialColor As String) As String

[JScript] public static function BuildColor(component : IComponent, owner : Control, initialColor : String) : String;

*Description*

Launches a color editor to build a color.

**Return Value:** The color value, represented as a string in an HTML color format, or **null** if the builder service could not be retrieved.

The returned string, if other than **null**, indicates a color in a valid HTML color format. Valid formats include named colors and color codes in RGB format (#RRGGBB). The component whose site is to be used to access design-time services. The control used to parent the picker window. The initial color to be shown in the picker window.

ControlDesigner class (System.Web.UI.Design)

ToString

#### *Description*

Provides a base class for Web server control designers.

ControlDesigner

*Example Syntax:*

ToString

[C#] public ControlDesigner();

[C++] public: ControlDesigner();

[VB] Public Sub New()

[JScript] public function ControlDesigner();

Properties:

AllowResize

ToString

[C#] public virtual bool AllowResize {get;}

[C++] public: \_\_property virtual bool get\_AllowResize();

1 [VB] Overridable Public ReadOnly Property AllowResize As Boolean

2 [JScript] public function get AllowResize() : Boolean;

4 *Description*

5 Gets a value indicating whether or not the control can be resized.

6 AssociatedComponents

7 Behavior

8 Component

9 DataBindings

10 DesignTimeElement

11 DesignTimeElementView

12 ToString

15 *Description*

16 Gets the designer view control object for the designer.

17 DesignTimeHtmlRequiresLoadComplete

18 ToString

20 [C#] public virtual bool DesignTimeHtmlRequiresLoadComplete {get;}

21 [C++] public: \_\_property virtual bool

22 get\_DesignTimeHtmlRequiresLoadComplete();

23 [VB] Overridable Public ReadOnly Property

24 DesignTimeHtmlRequiresLoadComplete As Boolean

25 [JScript] public function get DesignTimeHtmlRequiresLoadComplete() : Boolean;



1  
2 *Description*

3 Gets a value indicating whether the designer must have completed loading  
4 before the design time HTML can be used.

5 The default implementation returns **false** .

6 ID

7 ToString

8  
9 [C#] public virtual string ID {get; set;}

10 [C++] public: \_\_property virtual String\* get\_ID();public: \_\_property virtual void  
11 set\_ID(String\*);

12 [VB] Overridable Public Property ID As String

13 [JScript] public function get ID() : String;public function set ID(String);

14  
15 *Description*

16 Gets or sets the ID for the control designer.

17 InheritanceAttribute

18 Inherited

19 IsDirty

20 ToString

21  
22  
23 *Description*

24 Gets or sets a value indicating whether the Web server control has been  
25 marked as changed.

1       ReadOnly  
2       ToString  
3  
4   [C#] public bool ReadOnly {get; set;}  
5   [C++] public: \_\_property bool get\_ReadOnly();public: \_\_property void  
6   set\_ReadOnly(bool);  
7   [VB] Public Property ReadOnly As Boolean  
8   [JScript] public function get ReadOnly() : Boolean;public function set  
9   ReadOnly(Boolean);  
10  
11   *Description*  
12       Gets or sets a value indicating whether the control's associated design  
13   surface is set to read-only.  
14       ShadowProperties  
15       ShouldCodeSerialize  
16       Verbs  
17       CreatePlaceholderDesignTimeHtml  
18  
19   [C#] protected string CreatePlaceholderDesignTimeHtml();  
20   [C++] protected: String\* CreatePlaceholderDesignTimeHtml();  
21   [VB] Protected Function CreatePlaceholderDesignTimeHtml() As String  
22   [JScript] protected function CreatePlaceholderDesignTimeHtml() : String; Creates  
23   a simple HTML section for the control that can be used to display information  
24   related to the control at design-time when there is not enough information to  
25   display a meaningful representation of the control.

## Description

Creates a simple HTML section for the control that can be used to display information related to the control at design-time when there is not enough information to display a meaningful representation of the control.

*Return Value:* A string containing place-holder design-time HTML.

This method returns a string containing HTML in a standard place-holder style that an empty control can display at design-time to provide some basic information about the control. The string that is returned contains the type of the control and its ID.

### CreatePlaceHolderDesignTimeHtml

[C#] protected string CreatePlaceHolderDesignTimeHtml(string instruction);  
[C++] protected: String\* CreatePlaceHolderDesignTimeHtml(String\* instruction);  
[VB] Protected Function CreatePlaceHolderDesignTimeHtml(ByVal instruction As String) As String  
[JScript] protected function CreatePlaceHolderDesignTimeHtml(instruction : String) : String;

## Description

Creates a simple HTML section for the control, using the specified additional information, that can be used to display information related to the control at design-time when there is not enough information to display a meaningful representation of the control.

1 *Return Value:* A string containing place holder design-time HTML and the  
2 specified string.

3 This method returns a string containing HTML in a standard place-holder  
4 style that an empty control can display at design-time to provide some basic  
5 information about the control. The string that is returned contains the type of the  
6 control and its ID. A string conatining information to add to the HTML section.

#### 7 GetDesignTimeHtml

8  
9 [C#] public virtual string GetDesignTimeHtml();

10 [C++] public: virtual String\* GetDesignTimeHtml();

11 [VB] Overridable Public Function GetDesignTimeHtml() As String

12 [JScript] public function GetDesignTimeHtml() : String;

#### 13 14 *Description*

15 Gets the HTML to be used for the design-time representation of the control.

16 *Return Value:* The design-time HTML for the control.

#### 17 GetEmptyDesignTimeHtml

18  
19 [C#] protected virtual string GetEmptyDesignTimeHtml();

20 [C++] protected: virtual String\* GetEmptyDesignTimeHtml();

21 [VB] Overridable Protected Function GetEmptyDesignTimeHtml() As String

22 [JScript] protected function GetEmptyDesignTimeHtml() : String;

#### 23 24 *Description*

1 Gets the HTML to be used for the design-time representation of a blank  
2 control.

3 *Return Value:* The HTML used for the design-time representation of a blank  
4 control. This is the name of the component, by default.

5 The default behavior is to return a string containing the name of the  
6 component. This method is called by  
7 **System.Web.UI.Design.ControlDesigner.GetDesignTimeHtml** when there is no  
8 design-time HTML.

#### 9 GetErrorDesignTimeHtml

10  
11 [C#] protected virtual string GetErrorDesignTimeHtml(Exception e);

12 [C++] protected: virtual String\* GetErrorDesignTimeHtml(Exception\* e);

13 [VB] Overridable Protected Function GetErrorDesignTimeHtml(ByVal e As  
14 Exception) As String

15 [JScript] protected function GetErrorDesignTimeHtml(e : Exception) : String;

#### 16 17 *Description*

18 Gets the HTML to be used for the design-time representation of the control  
19 after an error has been encountered.

20 *Return Value:* The HTML for the specified exception. The exception that  
21 occurred.

#### 22 GetPersistInnerHtml

23  
24 [C#] public virtual string GetPersistInnerHtml();

25 [C++] public: virtual String\* GetPersistInnerHtml();

1 [VB] Overridable Public Function GetPersistInnerHtml() As String

2 [JScript] public function GetPersistInnerHtml() : String;

3  
4 *Description*

5 Gets the persistable inner HTML.

6 *Return Value:* The persistable inner HTML.

7 Initialize

8  
9 [C#] public override void Initialize(IComponent component);

10 [C++] public: void Initialize(IComponent\* component);

11 [VB] Overrides Public Sub Initialize(ByVal component As IComponent)

12 [JScript] public override function Initialize(component : IComponent);

13  
14 *Description*

15 Initializes the designer with the specified component.

16 This method is called by the designer host to establish the component being  
17 designed. The control element being designed.

18 IsPropertyBound

19  
20 [C#] public bool IsPropertyBound(string propName);

21 [C++] public: bool IsPropertyBound(String\* propName);

22 [VB] Public Function IsPropertyBound(ByVal propName As String) As Boolean

23 [JScript] public function IsPropertyBound(propName : String) : Boolean;

24  
25 *Description*

1 Gets a value indicating whether the specified property is data-bound.  
2 *Return Value:* **true** if the property is data bound; othwerise, **false** . The property to  
3 check for data binding.

#### 4 OnBehaviorAttached

5  
6 [C#] protected override void OnBehaviorAttached();  
7 [C++] protected: void OnBehaviorAttached();  
8 [VB] Overrides Protected Sub OnBehaviorAttached()  
9 [JScript] protected override function OnBehaviorAttached();  
10

#### 11 *Description*

12 Called when the designer is attached to the behavior.

#### 13 OnBindingsCollectionChanged

14  
15 [C#] protected override void OnBindingsCollectionChanged(string propName);  
16 [C++] protected: void OnBindingsCollectionChanged(String\* propName);  
17 [VB] Overrides Protected Sub OnBindingsCollectionChanged(ByVal propName  
18 As String)  
19 [JScript] protected override function OnBindingsCollectionChanged(propName :  
20 String);  
21

#### 22 *Description*

23 Called when the data bindings collection changes.  
24  
25

This method is called when the bindings collection has been changed by an external caller. In Visual Studio.NET, this method is called by the WebFormDataTable. The property to test for changes in its bindings collection.

#### OnComponentChanged

```
[C#] public virtual void OnComponentChanged(object sender,
ComponentChangedEventArgs ce);
[C++] public: virtual void OnComponentChanged(Object* sender,
ComponentChangedEventArgs* ce);
[VB] Overridable Public Sub OnComponentChanged(ByVal sender As Object,
ByVal ce As ComponentChangedEventArgs)
[JScript] public function OnComponentChanged(sender : Object, ce :
ComponentChangedEventArgs);
```

#### *Description*

Called when the component changes.

This method is called when a property is changed. It allows the implementer to do any processing that may be needed after a property change. The object that is the source of the event. A **System.ComponentModel.Design.ComponentChangedEventArgs** object that provides data about the event.

#### OnControlResize

```
[C#] protected virtual void OnControlResize();
[C++] protected: virtual void OnControlResize();
```



1 [VB] Overridable Protected Sub OnControlResize()

2 [JScript] protected function OnControlResize();

3  
4 *Description*

5 Called when the design-time control has been resized.

6 This method is typically only called by the design-time environment when  
7 a user action causes the control to be resized. This method may be called several  
8 times during a resize process to display the updated size of the control before the  
9 resize process is completed. The width and height properties of the control are  
10 updated before this method is called.

11 PreFilterProperties

12  
13 [C#] protected override void PreFilterProperties(IDictionary properties);

14 [C++] protected: void PreFilterProperties(IDictionary\* properties);

15 [VB] Overrides Protected Sub PreFilterProperties(ByVal properties As  
16 IDictionary)

17 [JScript] protected override function PreFilterProperties(properties : IDictionary);

18 RaiseResizeEvent

19  
20 [C#] public void RaiseResizeEvent();

21 [C++] public: void RaiseResizeEvent();

22 [VB] Public Sub RaiseResizeEvent()

23 [JScript] public function RaiseResizeEvent();

24  
25 *Description*

1       Raises the **System.Web.UI.Design.ControlDesigner.OnControlResize**  
2 event.

3       UpdateDesignTimeHtml

4  
5    [C#] public virtual void UpdateDesignTimeHtml();

6    [C++] public: virtual void UpdateDesignTimeHtml();

7    [VB] Overridable Public Sub UpdateDesignTimeHtml()

8    [JScript] public function UpdateDesignTimeHtml();

9  
10    *Description*

11       Updates the design-time HTML.

12       ControlParser class (System.Web.UI.Design)

13       UpdateDesignTimeHtml

14  
15  
16    *Description*

17       Provides methods for parsing the code for a control.

18       ParseControl

19  
20    [C#] public static Control ParseControl(IDesignerHost designerHost, string  
21    controlText);

22    [C++] public: static Control\* ParseControl(IDesignerHost\* designerHost, String\*  
23    controlText);

24    [VB] Public Shared Function ParseControl(ByVal designerHost As  
25    IDesignerHost, ByVal controlText As String) As Control

1 [JScript] public static function ParseControl(designerHost : IDesignerHost,  
2 controlText : String) : Control; Initializes a new instance of the  
3 **System.Web.UI.Design.ControlParser** class using the specified designer host  
4 and control text.

5  
6 *Description*

7       Initializes a new instance of the **System.Web.UI.Design.ControlParser**  
8 class using the specified designer host and control text.

9 *Return Value:* The Control that the specified text represents, or **null** if the parser  
10 could not build the control. The designer host for the document. The text of the  
11 code for the control.

12       ParseControl

13  
14 [C#] public static Control ParseControl(IDesignerHost designerHost, string  
15 controlText, string directives);

16 [C++] public: static Control\* ParseControl(IDesignerHost\* designerHost, String\*  
17 controlText, String\* directives);

18 [VB] Public Shared Function ParseControl(ByVal designerHost As  
19 IDesignerHost, ByVal controlText As String, ByVal directives As String) As  
20 Control

21 [JScript] public static function ParseControl(designerHost : IDesignerHost,  
22 controlText : String, directives : String) : Control;

23  
24 *Description*

25

1        Initializes a new instance of the **System.Web.UI.Design.ControlParser**  
2 class using the specified designer host, control text and directives.

3    *Return Value:* The Control that the specified text represents, or **null** if the parser  
4 could not build the control. The designer host of the document. The text of the  
5 code for the control. The directives to include in the code for the control.

#### 6        ParseTemplate

7  
8    [C#] public static ITemplate ParseTemplate(IDesignerHost designerHost, string  
9    templateText);

10    [C++] public: static ITemplate\* ParseTemplate(IDesignerHost\* designerHost,  
11    String\* templateText);

12    [VB] Public Shared Function ParseTemplate(ByVal designerHost As  
13    IDesignerHost, ByVal templateText As String) As ITemplate

14    [JScript] public static function ParseTemplate(designerHost : IDesignerHost,  
15    templateText : String) : ITemplate; Parses the specified text of a template.

#### 16 17    *Description*

18        Parses the specified text of a template.

19    *Return Value:* A new template based on the specified text. The designer host of  
20 the document. The text of the code for a template.

#### 21        ParseTemplate

22  
23    [C#] public static ITemplate ParseTemplate(IDesignerHost designerHost, string  
24    templateText, string directives);

25    [C++] public: static ITemplate\* ParseTemplate(IDesignerHost\* designerHost,

```

1 String* templateText, String* directives);
2 [VB] Public Shared Function ParseTemplate(ByVal designerHost As
3 IDesignerHost, ByVal templateText As String, ByVal directives As String) As
4 ITemplate
5 [JScript] public static function ParseTemplate(designerHost : IDesignerHost,
6 templateText : String, directives : String) : ITemplate;
7

```

#### *Description*

Parses the specified text of a template.

*Return Value:* A new template based on the specified text. The designer host of the document. The text of the code for a template. Any directives to add to the beginning of the code for the template.

ControlPersister class (System.Web.UI.Design)

ToString

#### *Description*

Provides helper functions for persisting Web server controls.

PersistControl

```

21 [C#] public static string PersistControl(Control control);
22 [C++] public: static String* PersistControl(Control* control);
23 [VB] Public Shared Function PersistControl(ByVal control As Control) As String
24 [JScript] public static function PersistControl(control : Control) : String; Gets a
25 string of persistence data that can persist a control.

```

1  
2 *Description*

3 Gets a string of persistence data that can persist a control.

4 *Return Value:* A string that contains the information to persist about the control.

5 The **System.Web.UI.Control** to persist.

6 PersistControl

7  
8 [C#] public static string PersistControl(Control control, IDesignerHost host);

9 [C++] public: static String\* PersistControl(Control\* control, IDesignerHost\*  
10 host);

11 [VB] Public Shared Function PersistControl(ByVal control As Control, ByVal  
12 host As IDesignerHost) As String

13 [JScript] public static function PersistControl(control : Control, host :  
14 IDesignerHost) : String;

15  
16 *Description*

17 Gets a string of persistence data that can persist a control.

18 *Return Value:* A string that contains the information to persist about the control.

19 The **System.Web.UI.Control** to persist. The designer host for the control.

20 PersistControl

21  
22 [C#] public static void PersistControl(TextWriter sw, Control control);

23 [C++] public: static void PersistControl(TextWriter\* sw, Control\* control);

24 [VB] Public Shared Sub PersistControl(ByVal sw As TextWriter, ByVal control  
25 As Control)

1 [JScript] public static function PersistControl(sw : TextWriter, control : Control);

2  
3 *Description*

4 Persists a control using the specified text writer. The  
5 **System.IO.TextWriter** to use. The **System.Web.UI.Control** to persist.

6 PersistControl

7  
8 [C#] public static void PersistControl(TextWriter sw, Control control,  
9 IDesignerHost host);

10 [C++] public: static void PersistControl(TextWriter\* sw, Control\* control,  
11 IDesignerHost\* host);

12 [VB] Public Shared Sub PersistControl(ByVal sw As TextWriter, ByVal control  
13 As Control, ByVal host As IDesignerHost)

14 [JScript] public static function PersistControl(sw : TextWriter, control : Control,  
15 host : IDesignerHost);

16  
17 *Description*

18 Persists a control using the specified text writer. The  
19 **System.IO.TextWriter** to use. The **System.Web.UI.Control** to persist. The  
20 designer host for the control.

21 PersistInnerProperties

22  
23 [C#] public static string PersistInnerProperties(object component, IDesignerHost  
24 host);

25 [C++] public: static String\* PersistInnerProperties(Object\* component,

```

1 IDesignerHost* host);
2 [VB] Public Shared Function PersistInnerProperties(ByVal component As Object,
3 ByVal host As IDesignerHost) As String
4 [JScript] public static function PersistInnerProperties(component : Object, host :
5 IDesignerHost) : String; Gets a string of persistence data that can persist the inner
6 properties of a control.

```

#### *Description*

Gets a string of persistence data that can persist the inner properties of a control.

*Return Value:* A string that contains the information to persist about the inner properties of the control. The **System.Web.UI.Control** to persist. The designer host of the control.

#### **PersistInnerProperties**

```

16 [C#] public static void PersistInnerProperties(TextWriter sw, object component,
17 IDesignerHost host);
18 [C++] public: static void PersistInnerProperties(TextWriter* sw, Object*
19 component, IDesignerHost* host);
20 [VB] Public Shared Sub PersistInnerProperties(ByVal sw As TextWriter, ByVal
21 component As Object, ByVal host As IDesignerHost)
22 [JScript] public static function PersistInnerProperties(sw : TextWriter, component
23 : Object, host : IDesignerHost);

```

#### *Description*



Persists the inner properties of the control using the specified text writer.  
The **System.IO.TextWriter** to use. The **System.Web.UI.Control** to persist. The  
designer host of the control.

DataBindingCollectionConverter class (System.Web.UI.Design)  
ToString

#### *Description*

Provides a type converter to convert data binding collections to various  
other representations.

DataBindingCollectionConverter

#### *Example Syntax:*

ToString

[C#] public DataBindingCollectionConverter();

[C++] public: DataBindingCollectionConverter();

[VB] Public Sub New()

[JScript] public function DataBindingCollectionConverter();

ConvertTo

[C#] public override object ConvertTo(ITypeDescriptorContext context,

CultureInfo culture, object value, Type destinationType);

[C++] public: Object\* ConvertTo(ITypeDescriptorContext\* context, CultureInfo\*

culture, Object\* value, Type\* destinationType);

[VB] Overrides Public Function ConvertTo(ByVal context As

1 ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object,  
2 ByVal destinationType As Type) As Object  
3 [JScript] public override function ConvertTo(context : ITypeDescriptorContext,  
4 culture : CultureInfo, value : Object, destinationType : Type) : Object;

5  
6 *Description*

7 Converts a data binding collection to the specified type.

8 *Return Value:* The object produced by the type conversion. An

9 **System.ComponentModel.ITypeDescriptorContext** that indicates the  
10 component or control the data binding collection belongs to. A  
11 **System.Globalization.CultureInfo** that can be used to provide additional culture  
12 information. The object to convert. The type to convert to.

13 DataBindingCollectionEditor class (System.Web.UI.Design)

14 ToString

15  
16  
17 *Description*

18 Provides user interface for editing a collection of data bindings.

19 DataBindingCollectionEditor

20 *Example Syntax:*

21 ToString

22  
23 [C#] public DataBindingCollectionEditor();

24 [C++] public: DataBindingCollectionEditor();  
25

1 [VB] Public Sub New()

2 [JScript] public function DataBindingCollectionEditor();

3     EditValue

5 [C#] public override object EditValue(ITypeDescriptorContext context,

6 IServiceProvider provider, object value);

7 [C++] public: Object\* EditValue(ITypeDescriptorContext\* context,

8 IServiceProvider\* provider, Object\* value);

9 [VB] Overrides Public Function EditValue(ByVal context As

10 ITypeDescriptorContext, ByVal provider As IServiceProvider, ByVal value As

11 Object) As Object

12 [JScript] public override function EditValue(context : ITypeDescriptorContext,

13 provider : IServiceProvider, value : Object) : Object;

15 *Description*

16     Edits the value of the specified data binding collection using the specified  
17 service provider and context.

18 *Return Value:* The new collection. An

19 **System.ComponentModel.ITypeDescriptorContext** that identifies the  
20 component or control the collection belongs to. The **System.IServiceProvider** to  
21 use. The collection to edit.

22     GetEditStyle

24 [C#] public override UITypeEditorEditStyle

25 GetEditStyle(ITypeDescriptorContext context);

```

1 [C++] public: UITypeEditorEditStyle GetEditStyle(ITypeDescriptorContext*
2 context);
3 [VB] Overrides Public Function GetEditStyle(ByVal context As
4 ITypeDescriptorContext) As UITypeEditorEditStyle
5 [JScript] public override function GetEditStyle(context : ITypeDescriptorContext)
6 : UITypeEditorEditStyle;
7

```

### *Description*

Gets the editor style used by the **System.Web.UI.Design.DataBindingCollectionEditor.EditValue(System.ComponentModel.ITypeDescriptorContext, System.IServiceProvider, System.Object)** method.

**Return Value:** A **System.Drawing.Design.UITypeEditorEditStyle** that specifies the editor edit style of the component or control. An **System.ComponentModel.ITypeDescriptorContext** that identifies the component or control to retrieve the edit style for.

DataBindingHandler class (System.Web.UI.Design)

ToString

### *Description*

Provides a base class for a data binding handler.

DataBindingHandler

*Example Syntax:*

ToString

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#] protected DataBindingHandler();
[C++] protected: DataBindingHandler();
[VB] Protected Sub New()
[JScript] protected function DataBindingHandler();

    DataBindControl

[C#] public abstract void DataBindControl(IDesignerHost designerHost, Control
control);
[C++] public: virtual void DataBindControl(IDesignerHost* designerHost,
Control* control) = 0;
[VB] MustOverride Public Sub DataBindControl(ByVal designerHost As
IDesignerHost, ByVal control As Control)
[JScript] public abstract function DataBindControl(designerHost : IDesignerHost,
control : Control);
```

*Description*

Adds this data binding to the specified control. The designer host of the document. The control to data bind to.

DataBindingValueUIHandler class (System.Web.UI.Design)  
ToString

*Description*

Provides a UI handler for data binding values.

1       DataBindingValueUIHandler

2       *Example Syntax:*

3       ToString

4  
5    [C#] public DataBindingValueUIHandler();

6    [C++] public: DataBindingValueUIHandler();

7    [VB] Public Sub New()

8    [JScript] public function DataBindingValueUIHandler();

9       OnGetUIValueItem

10  
11   [C#] public void OnGetUIValueItem(ITypeDescriptorContext context,

12   PropertyDescriptor propDesc, ArrayList valueUIItemList);

13   [C++] public: void OnGetUIValueItem(ITypeDescriptorContext\* context,

14   PropertyDescriptor\* propDesc, ArrayList\* valueUIItemList);

15   [VB] Public Sub OnGetUIValueItem(ByVal context As ITypeDescriptorContext,

16   ByVal propDesc As PropertyDescriptor, ByVal valueUIItemList As ArrayList)

17   [JScript] public function OnGetUIValueItem(context : ITypeDescriptorContext,

18   propDesc : PropertyDescriptor, valueUIItemList : ArrayList);

19  
20    *Description*

21       Adds a data binding for the specified property and the specified value item  
22   list if the current control has data bindings and the current object does not already  
23   have a binding. An **System.ComponentModel.ITypeDescriptorContext** object  
24   that can provide additional context information. A

**System.ComponentModel.PropertyDescriptor** that represents the property to add a data binding for. A list of items that have data bindings.

DataFieldConverter class (System.Web.UI.Design)

ToString

### *Description*

Provides a type converter that can retrieve a list of data fields that are accessible through the current component's selected data source, and convert a data field name to string.

**System.Web.UI.Design.DataFieldConverter** provides methods that can be used to do the following: Convert a data field name string to string.

DataFieldConverter

### *Example Syntax:*

ToString

[C#] public DataFieldConverter();

[C++] public: DataFieldConverter();

[VB] Public Sub New()

[JScript] public function DataFieldConverter();

### *Description*

Initializes a new instance of the

**System.Web.UI.Design.DataFieldConverter** class.

CanConvertFrom

```

1
2 [C#] public override bool CanConvertFrom(ITypeDescriptorContext context,
3     Type sourceType);
4 [C++] public: bool CanConvertFrom(ITypeDescriptorContext* context, Type*
5     sourceType);
6 [VB] Overrides Public Function CanConvertFrom(ByVal context As
7     ITypeDescriptorContext, ByVal sourceType As Type) As Boolean
8 [JScript] public override function CanConvertFrom(context :
9     ITypeDescriptorContext, sourceType : Type) : Boolean;
10

```

### *Description*

Gets a value indicating whether the converter can convert an object of the specified source type to the native type of the converter.

**Return Value:** **true** if the converter can perform the conversion; otherwise, **false**.

This method returns **true** if the source type is a string. Otherwise, this method always returns **false**. An **System.ComponentModel.ITypeDescriptorContext** that can be used to gain additional context information. A **System.Type** that represents the type you wish to convert from.

### **ConvertFrom**

```

21
22 [C#] public override object ConvertFrom(ITypeDescriptorContext context,
23     CultureInfo culture, object value);
24 [C++] public: Object* ConvertFrom(ITypeDescriptorContext* context,
25     CultureInfo* culture, Object* value);

```



```

1 [VB] Overrides Public Function ConvertFrom(ByVal context As
2 ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object)
3 As Object
4 [JScript] public override function ConvertFrom(context : ITypeDescriptorContext,
5 culture : CultureInfo, value : Object) : Object;

```

### *Description*

Converts the specified object to the native type of the converter.

*Return Value:* An **System.Object** that represents the specified object after conversion.

Override this method to provide your own conversion requirements. An **System.ComponentModel.ITypeDescriptorContext** that can be used to gain additional context information. A **System.Globalization.CultureInfo** that can be used to support localization features. The **System.Object** to convert.

### *GetStandardValues*

```

17 [C#] public override StandardValuesCollection
18 GetStandardValues(ITypeDescriptorContext context);
19 [C++] public: StandardValuesCollection*
20 GetStandardValues(ITypeDescriptorContext* context);
21 [VB] Overrides Public Function GetStandardValues(ByVal context As
22 ITypeDescriptorContext) As StandardValuesCollection
23 [JScript] public override function GetStandardValues(context :
24 ITypeDescriptorContext) : StandardValuesCollection;

```

1  
2 *Description*

3 Gets the data fields present within the selected data source if information  
4 about them is available.

5 *Return Value:* A

6 **System.ComponentModel.TypeConverter.StandardValuesCollection** listing  
7 the standard accessible data sources. An  
8 **System.ComponentModel.ITypeDescriptorContext** indicating the component  
9 or control to get values for.

10 GetStandardValuesExclusive

11  
12 [C#] public override bool GetStandardValuesExclusive(ITypeDescriptorContext  
13 context);

14 [C++] public: bool GetStandardValuesExclusive(ITypeDescriptorContext\*  
15 context);

16 [VB] Overrides Public Function GetStandardValuesExclusive(ByVal context As  
17 ITypeDescriptorContext) As Boolean

18 [JScript] public override function GetStandardValuesExclusive(context :  
19 ITypeDescriptorContext) : Boolean;

20  
21 *Description*

22 Gets a value indicating whether the collection of standard values returned  
23 from **System.ComponentModel.TypeConverter.GetStandardValues** is an  
24 exclusive list of all possible values.

25 *Return Value:* **true** if the

1 **System.ComponentModel.TypeConverter.StandardValuesCollection** returned  
2 from **System.ComponentModel.TypeConverter.GetStandardValues** is an all  
3 exclusive list of all possible values; **false** if other values are possible.

4 If the list is exclusive, such as in an enumeration data type, then no other  
5 values are valid. If the list is not exclusive, then there are other valid values  
6 besides the list of standard values that

7 **System.ComponentModel.TypeConverter.GetStandardValues** provides. An  
8 **System.ComponentModel.ITypeDescriptorContext** that can be used to gain  
9 additional context information.

10 **GetStandardValuesSupported**

11  
12 [C#] public override bool GetStandardValuesSupported(ITypeDescriptorContext  
13 context);

14 [C++] public: bool GetStandardValuesSupported(ITypeDescriptorContext\*  
15 context);

16 [VB] Overrides Public Function GetStandardValuesSupported(ByVal context As  
17 ITypeDescriptorContext) As Boolean

18 [JScript] public override function GetStandardValuesSupported(context :  
19 ITypeDescriptorContext) : Boolean; Gets a value indicating whether the converter  
20 supports a standard set of values that can be picked from a list.

21  
22 *Description*

23 Gets a value indicating whether the converter supports a standard set of  
24 values that can be picked from a list.

25 *Return Value:* **true** if

1 **System.ComponentModel.TypeConverter.GetStandardValues** can be called to  
2 find a common set of values the object supports; otherwise, **false** . An  
3 **System.ComponentModel.ITypeDescriptorContext** that can be used to gain  
4 additional context information.

5       DataMemberConverter class (System.Web.UI.Design)

6       ToString

7  
8  
9 *Description*

10       Provides a type converter that can retrieve a list of data members that are  
11 accessible through the current component's selected data source, and convert a  
12 data member name to string.

13       **System.Web.UI.Design.DataMemberConverter** provides methods that  
14 can do the following: Convert a data member name string to string.

15       DataMemberConverter

16 *Example Syntax:*

17       ToString

18  
19 [C#] public DataMemberConverter();

20 [C++] public: DataMemberConverter();

21 [VB] Public Sub New()

22 [JScript] public function DataMemberConverter();

23  
24 *Description*

1        Initializes a new instance of the  
2   **System.Web.UI.Design.DataFieldConverter** class.

3        CanConvertFrom

4  
5   [C#] public override bool CanConvertFrom(ITypeDescriptorContext context,  
6   Type sourceType);  
7   [C++] public: bool CanConvertFrom(ITypeDescriptorContext\* context, Type\*  
8   sourceType);  
9   [VB] Overrides Public Function CanConvertFrom(ByVal context As  
10   ITypeDescriptorContext, ByVal sourceType As Type) As Boolean  
11   [JScript] public override function CanConvertFrom(context :  
12   ITypeDescriptorContext, sourceType : Type) : Boolean;

13  
14   *Description*

15        Gets a value indicating whether the converter can convert an object of the  
16   specified source type to the native type of the converter.

17   *Return Value:* **true** if the converter can perform the conversion; otherwise, **false** .

18        Override this method to provide your own conversion requirements. An  
19   **System.ComponentModel.ITypeDescriptorContext** that can be used to gain  
20   additional context information. A **System.Type** that represents the type you wish  
21   to convert from.

22        ConvertFrom

23  
24   [C#] public override object ConvertFrom(ITypeDescriptorContext context,  
25   CultureInfo culture, object value);

```

1 [C++] public: Object* ConvertFrom(ITypeDescriptorContext* context,
2 CultureInfo* culture, Object* value);
3 [VB] Overrides Public Function ConvertFrom(ByVal context As
4 ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object)
5 As Object
6 [JScript] public override function ConvertFrom(context : ITypeDescriptorContext,
7 culture : CultureInfo, value : Object) : Object;
8

```

### *Description*

Converts the specified object to the native type of the converter.

*Return Value:* An **System.Object** that represents the specified object after conversion.

Override this method to provide your own conversion requirements. An **System.ComponentModel.ITypeDescriptorContext** that can be used to gain additional context information. A **System.Globalization.CultureInfo** that can be used to support localization features. The **System.Object** to convert.

### *GetStandardValues*

```

19 [C#] public override StandardValuesCollection
20 GetStandardValues(ITypeDescriptorContext context);
21 [C++] public: StandardValuesCollection*
22 GetStandardValues(ITypeDescriptorContext* context);
23 [VB] Overrides Public Function GetStandardValues(ByVal context As
24 ITypeDescriptorContext) As StandardValuesCollection
25 [JScript] public override function GetStandardValues(context :

```

1 ITypeDescriptorContext) : StandardValuesCollection;

2  
3 *Description*

4 Gets the data members present within the selected data source, if  
5 information about them is available.

6 *Return Value:* A

7 **System.ComponentModel.TypeConverter.StandardValuesCollection** listing  
8 the standard accessible data sources. An

9 **System.ComponentModel.ITypeDescriptorContext** indicating the component  
10 or control to get values for.

11 .GetStandardValuesExclusive

12  
13 [C#] public override bool GetStandardValuesExclusive(ITypeDescriptorContext  
14 context);

15 [C++] public: bool GetStandardValuesExclusive(ITypeDescriptorContext\*  
16 context);

17 [VB] Overrides Public Function GetStandardValuesExclusive(ByVal context As  
18 ITypeDescriptorContext) As Boolean

19 [JScript] public override function GetStandardValuesExclusive(context :  
20 ITypeDescriptorContext) : Boolean;

21  
22 *Description*

23 Gets a value indicating whether the collection of standard values returned  
24 from **System.ComponentModel.TypeConverter.GetStandardValues** is an  
25 exclusive list of all possible values.

1 *Return Value:* **true** if the

2 **System.ComponentModel.TypeConverter.StandardValuesCollection** returned  
3 from **System.ComponentModel.TypeConverter.GetStandardValues** is an  
4 exclusive list of possible values; **false** if other values are possible.

5 If the list is exclusive, such as in an enumeration data type, then no other  
6 values are valid. If the list is not exclusive, then there are other valid values  
7 besides the list of standard values that

8 **System.ComponentModel.TypeConverter.GetStandardValues** provides. An  
9 **System.ComponentModel.ITypeDescriptorContext** that provides a format  
10 context.

11 **GetStandardValuesSupported**

12  
13 [C#] public override bool GetStandardValuesSupported(ITypeDescriptorContext  
14 context);

15 [C++] public: bool GetStandardValuesSupported(ITypeDescriptorContext\*  
16 context);

17 [VB] Overrides Public Function GetStandardValuesSupported(ByVal context As  
18 ITypeDescriptorContext) As Boolean

19 [JScript] public override function GetStandardValuesSupported(context :  
20 ITypeDescriptorContext) : Boolean; Gets a value indicating whether the converter  
21 supports a standard set of values that can be picked from a list.

22  
23 *Description*

24 Gets a value indicating whether the converter supports a standard set of  
25 values that can be picked from a list.



1 *Return Value:* **true** if

2 **System.ComponentModel.TypeConverter.GetStandardValues** should be  
3 called to find a common set of values the object supports; otherwise, **false** . An  
4 **System.ComponentModel.ITypeDescriptorContext** that can be used to gain  
5 additional context information.

6       DataSourceConverter class (System.Web.UI.Design)

7       ToString

8  
9  
10 *Description*

11       Provides a type converter that can retrieve a list of data sources accessible  
12 to the current component, and convert a data source name to string.

13       **System.Web.UI.Design.DataSourceConverter** provides methods that can  
14 be used to do the following: Convert a data source name string to string.

15       DataSourceConverter

16 *Example Syntax:*

17       ToString

18  
19 [C#] public DataSourceConverter();

20 [C++] public: DataSourceConverter();

21 [VB] Public Sub New()

22 [JScript] public function DataSourceConverter();

23  
24 *Description*

1        Initializes a new instance of the  
2        **System.Web.UI.Design.DataSourceConverter** class.

3        CanConvertFrom

4  
5        [C#] public override bool CanConvertFrom(ITypeDescriptorContext context,  
6        Type sourceType);  
7        [C++] public: bool CanConvertFrom(ITypeDescriptorContext\* context, Type\*  
8        sourceType);  
9        [VB] Overrides Public Function CanConvertFrom(ByVal context As  
10        ITypeDescriptorContext, ByVal sourceType As Type) As Boolean  
11        [JScript] public override function CanConvertFrom(context :  
12        ITypeDescriptorContext, sourceType : Type) : Boolean;

13  
14        *Description*

15        Gets a value indicating whether the converter can convert an object of the  
16        specified source type to the native type of the converter.

17        *Return Value:* **true** if the converter can perform the conversion; otherwise, **false** .

18        Override this method to provide your own conversion requirements. An  
19        **System.ComponentModel.ITypeDescriptorContext** that can be used to gain  
20        additional context information. A **System.Type** that represents the type you wish  
21        to convert from.

22        ConvertFrom

23  
24        [C#] public override object ConvertFrom(ITypeDescriptorContext context,  
25        CultureInfo culture, object value);

```

1 [C++] public: Object* ConvertFrom(ITypeDescriptorContext* context,
2 CultureInfo* culture, Object* value);
3 [VB] Overrides Public Function ConvertFrom(ByVal context As
4 ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object)
5 As Object
6 [JScript] public override function ConvertFrom(context : ITypeDescriptorContext,
7 culture : CultureInfo, value : Object) : Object;
8

```

### *Description*

Converts the specified object to the native type of the converter.

*Return Value:* An **System.Object** that represents the specified object after conversion.

Override this method to provide your own conversion requirements. An **System.ComponentModel.ITypeDescriptorContext** that can be used to gain additional context information. The **System.Globalization.CultureInfo** that can be used to support localization features. The **System.Object** to convert.

### *GetStandardValues*

```

19 [C#] public override StandardValuesCollection
20 GetStandardValues(ITypeDescriptorContext context);
21 [C++] public: StandardValuesCollection*
22 GetStandardValues(ITypeDescriptorContext* context);
23 [VB] Overrides Public Function GetStandardValues(ByVal context As
24 ITypeDescriptorContext) As StandardValuesCollection
25 [JScript] public override function GetStandardValues(context :

```

1 ITypeDescriptorContext) : StandardValuesCollection;

2  
3 *Description*

4 Gets the standard data sources accessible to the control.

5 *Return Value:* A

6 **System.ComponentModel.TypeConverter.StandardValuesCollection** listing  
7 the standard accessible data sources. An

8 **System.ComponentModel.ITypeDescriptorContext** indicating the component  
9 or control to get values for.

10 GetStandardValuesExclusive

11  
12 [C#] public override bool GetStandardValuesExclusive(ITypeDescriptorContext  
13 context);

14 [C++] public: bool GetStandardValuesExclusive(ITypeDescriptorContext\*  
15 context);

16 [VB] Overrides Public Function GetStandardValuesExclusive(ByVal context As  
17 ITypeDescriptorContext) As Boolean

18 [JScript] public override function GetStandardValuesExclusive(context :  
19 ITypeDescriptorContext) : Boolean;

20  
21 *Description*

22 Gets a value indicating whether the collection of standard values returned  
23 from **System.ComponentModel.TypeConverter.GetStandardValues** is an  
24 exclusive list of all possible values.

25 *Return Value:* **true** if the

1 **System.ComponentModel.TypeConverter.StandardValuesCollection** returned  
2 from **System.ComponentModel.TypeConverter.GetStandardValues** is an  
3 exclusive list of all possible values; **false** if other values are possible.

4 If the list is exclusive, such as in an enumeration data type, then no other  
5 values are valid. If the list is not exclusive, then there are other valid values  
6 besides the list of standard values that

7 **System.ComponentModel.TypeConverter.GetStandardValues** provides. An  
8 **System.ComponentModel.ITypeDescriptorContext** that can be used to gain  
9 additional context information.

#### 10 **GetStandardValuesSupported**

11  
12 [C#] public override bool GetStandardValuesSupported(ITypeDescriptorContext  
13 context);

14 [C++] public: bool GetStandardValuesSupported(ITypeDescriptorContext\*  
15 context);

16 [VB] Overrides Public Function GetStandardValuesSupported(ByVal context As  
17 ITypeDescriptorContext) As Boolean

18 [JScript] public override function GetStandardValuesSupported(context :  
19 ITypeDescriptorContext) : Boolean; Gets a value indicating whether the converter  
20 supports a standard set of values that can be picked from a list.

#### 21 22 *Description*

23 Gets a value indicating whether the converter supports a standard set of  
24 values that can be picked from a list.

25 *Return Value:* **true** if

1 **System.ComponentModel.TypeConverter.GetStandardValues** should be  
2 called to find a common set of values the object supports; otherwise, **false** . An  
3 **System.ComponentModel.ITypeDescriptorContext** that can be used to gain  
4 additional context information.

5       DesignTimeData class (System.Web.UI.Design)

6       ToString

7  
8  
9 *Description*

10       Provides helper methods that can be used by control designers to generate  
11 sample data for use in design time databinding.

12       ToString

13  
14 [C#] public static readonly EventHandler DataBindingHandler;

15 [C++] public: static EventHandler\* DataBindingHandler;

16 [VB] Public Shared ReadOnly DataBindingHandler As EventHandler

17 [JScript] public static var DataBindingHandler : EventHandler;

18       CreateDummyDataTable

19  
20 [C#] public static DataTable CreateDummyDataTable();

21 [C++] public: static DataTable\* CreateDummyDataTable();

22 [VB] Public Shared Function CreateDummyDataTable() As DataTable

23 [JScript] public static function CreateDummyDataTable() : DataTable;

24  
25 *Description*

1       Creates a datatable that contains sample data.  
2   *Return Value:* A new **System.Data.DataTable** that contains 3 columns of type  
3   string.

4       CreateSampleDataTable

5  
6   [C#] public static DataTable CreateSampleDataTable(IEnumerable  
7   referenceData);

8   [C++] public: static DataTable\* CreateSampleDataTable(IEnumerable\*  
9   referenceData);

10   [VB] Public Shared Function CreateSampleDataTable(ByVal referenceData As  
11   IEnumerable) As DataTable

12   [JScript] public static function CreateSampleDataTable(referenceData :  
13   IEnumerable) : DataTable;

14  
15   *Description*

16       Creates a sample datatable with the same schema as the supplied  
17   datasource.

18   *Return Value:* A sample data table that contains dummy data in the format of the  
19   specified data source, or a default dummy data table if the specified data source  
20   contained no data columns. A data source with the schema to use as the format for  
21   the sample datatable.

22       GetDataFields

23  
24   [C#] public static PropertyDescriptorCollection GetDataFields(IEnumerable  
25   dataSource);

1 [C++] public: static PropertyDescriptorCollection\* GetDataFields(IEnumerable\*  
2 dataSource);

3 [VB] Public Shared Function GetDataFields(ByVal dataSource As IEnumerable)  
4 As PropertyDescriptorCollection

5 [JScript] public static function GetDataFields(dataSource : IEnumerable) :  
6 PropertyDescriptorCollection;

7  
8 *Description*

9 Gets the data fields of the specified data source.

10 *Return Value:* A **System.ComponentModel.PropertyDescriptorCollection** that  
11 represents the data fields of the specified data source. The data source to retrieve  
12 the data fields of.

13 **GetDataMember**

14  
15 [C#] public static IEnumerable GetDataMember(IListSource dataSource, string  
16 dataMember);

17 [C++] public: static IEnumerable\* GetDataMember(IListSource\* dataSource,  
18 String\* dataMember);

19 [VB] Public Shared Function GetDataMember(ByVal dataSource As IListSource,  
20 ByVal dataMember As String) As IEnumerable

21 [JScript] public static function GetDataMember(dataSource : IListSource,  
22 dataMember : String) : IEnumerable;

23  
24 *Description*



Gets the specified data member from the specified data source.

*Return Value:* The specified data member from the specified data source, if it exists.

This method searches the specified data source for the specified data member. If the *dataMember* property is **null**, the first item in the specified data source is returned. The data source that contains the member to retrieve. The data member to retrieve.

#### GetDataMembers

[C#] public static string[] GetDataMembers(object dataSource);

[C++] public: static String\* GetDataMembers(Object\* dataSource) \_\_gc[];

[VB] Public Shared Function GetDataMembers(ByVal dataSource As Object) As String()

[JScript] public static function GetDataMembers(dataSource : Object) : String[];

#### Description

Gets the data members of the specified data source.

*Return Value:* A string array that represents the data members of the specified data source. The data source to retrieve the members of.

#### GetDesignTimeDataSource

[C#] public static IEnumerable GetDesignTimeDataSource(DataTable dataTable, int minimumRows);

[C++] public: static IEnumerable\* GetDesignTimeDataSource(DataTable\* dataTable, int minimumRows);

1 [VB] Public Shared Function GetDesignTimeDataSource(ByVal dataTable As  
2 DataTable, ByVal minimumRows As Integer) As IEnumerable  
3 [JScript] public static function GetDesignTimeDataSource(dataTable : DataTable,  
4 minimumRows : int) : IEnumerable;

5  
6 *Description*

7 Adds sample rows to the specified data table and returns the updated table.

8 *Return Value:* A live data source for use at design time. The table that defines the  
9 shape of the data source. Sample rows are added to this table. The minimum  
10 number of rows of sample data that the data source will contain.

11 GetSelectedDataSource

12  
13 [C#] public static object GetSelectedDataSource(IComponent component, string  
14 dataSource);

15 [C++] public: static Object\* GetSelectedDataSource(IComponent\* component,  
16 String\* dataSource);

17 [VB] Public Shared Function GetSelectedDataSource(ByVal component As  
18 IComponent, ByVal dataSource As String) As Object

19 [JScript] public static function GetSelectedDataSource(component : IComponent,  
20 dataSource : String) : Object; Gets the specified data source.

21  
22 *Description*

23 Gets the specified data source.

24 *Return Value:* The data source object, or **null** if the data source, specified  
25

component's site, or the container of the data source could not be accessed. The component that contains the data source. The data source to retrieve.

### GetSelectedDataSource

```
[C#] public static IEnumerable GetSelectedDataSource(IComponent component,
string dataSource, string dataMember);
```

```
[C++] public: static IEnumerable* GetSelectedDataSource(IComponent*
component, String* dataSource, String* dataMember);
```

```
[VB] Public Shared Function GetSelectedDataSource(ByVal component As
IComponent, ByVal dataSource As String, ByVal dataMember As String) As
IEnumerable
```

```
[JScript] public static function GetSelectedDataSource(component : IComponent,
dataSource : String, dataMember : String) : IEnumerable;
```

### *Description*

Gets the specified data member of the specified data source.

**Return Value:** The data member, or **null** if the data source could not be accessed.

The component that contains the data source. The data source to retrieve. The data member to retrieve.

HtmlControlDesigner class (System.Web.UI.Design)

### ToString

### *Description*

Provides basic design-time functionality for ASP.NET server controls.

1       HtmlControlDesigner

2       *Example Syntax:*

3       ToString

4  
5       [C#] public HtmlControlDesigner();

6       [C++] public: HtmlControlDesigner();

7       [VB] Public Sub New()

8       [JScript] public function HtmlControlDesigner();

9  
10      *Description*

11       Initializes a new instance of the

12      **System.Web.UI.Design.HtmlControlDesigner** class.

13       AssociatedComponents

14       Behavior

15       ToString

16  
17  
18      *Description*

19       Gets or sets the DHTML behavior associated with the designer instance.

20       Component

21       DataBindings

22       ToString

23  
24  
25      *Description*

1 Gets or sets the data bindings for the current control.

2 DesignTimeElement

3 ToString

4  
5 [C#] protected object DesignTimeElement {get;}

6 [C++] protected: \_\_property Object\* get\_DesignTimeElement();

7 [VB] Protected ReadOnly Property DesignTimeElement As Object

8 [JScript] protected function get DesignTimeElement() : Object;

9  
10 *Description*

11 Gets the base Web Forms designer associated to the ASPX document.

12 InheritanceAttribute

13 Inherited

14 ShadowProperties

15 ShouldCodeSerialize

16 ToString

17  
18  
19 *Description*

20 Indicates whether a field declaration for the control should be created in the  
21 code-behind file for the current design document.

22 Verbs

23 Dispose

24  
25 [C#] protected override void Dispose(bool disposing);

1 [C++] protected: void Dispose(bool disposing);

2 [VB] Overrides Protected Sub Dispose(ByVal disposing As Boolean)

3 [JScript] protected override function Dispose(disposing : Boolean);

4  
5 *Description*

6 Releases the unmanaged resources used by the  
7 **System.Web.UI.Design.HtmlControlDesigner** and optionally releases the  
8 managed resources.

9 This method is called by the public **Dispose()** method and the  
10 **System.Object.Finalize** method. **true** to release both managed and unmanaged  
11 resources; **false** to release only unmanaged resources.

12  
13 *Description*

14 Initializes the designer and sets the component for design.

15 **System.Web.UI.Design.ControlDesigner.Initialize(System.Component**  
16 **Model.IComponent)** should be called by the designer host to initialize the  
17 designer. The control element for design.

18 OnBehaviorAttached

19  
20 [C#] protected virtual void OnBehaviorAttached();

21 [C++] protected: virtual void OnBehaviorAttached();

22 [VB] Overridable Protected Sub OnBehaviorAttached()

23 [JScript] protected function OnBehaviorAttached();

24  
25 *Description*

Notification that is called when a behavior is attached to the designer.

## OnBehaviorDetaching

[C#] protected virtual void OnBehaviorDetaching();

[C++] protected: virtual void OnBehaviorDetaching();

[VB] Overridable Protected Sub OnBehaviorDetaching()

[JScript] protected function OnBehaviorDetaching();

### *Description*

Notification that is called when a behavior is detached from the designer.

## OnBindingsCollectionChanged

[C#] protected virtual void OnBindingsCollectionChanged(string propName);

[C++] protected: virtual void OnBindingsCollectionChanged(String\* propName);

[VB] Overridable Protected Sub OnBindingsCollectionChanged(ByVal propName  
As String)

[JScript] protected function OnBindingsCollectionChanged(propName : String);

### *Description*

Provides a method that can be used to indicate when a data binding has been changed.

This method should be called by a control designer after it has made a change to the data bindings for the control. The name of the property that has been changed.

## OnSetParent

```

1
2 [C#] public virtual void OnSetParent();
3 [C++] public: virtual void OnSetParent();
4 [VB] Overridable Public Sub OnSetParent()
5 [JScript] public function OnSetParent();
6

```

### *Description*

Notification that is called when the associated control is parented.

### **PreFilterEvents**

```

11 [C#] protected override void PreFilterEvents(IDictionary events);
12 [C++] protected: void PreFilterEvents(IDictionary* events);
13 [VB] Overrides Protected Sub PreFilterEvents(ByVal events As IDictionary)
14 [JScript] protected override function PreFilterEvents(events : IDictionary);
15

```

### *Description*

Allows a designer to filter the set of member events that the component it is designing will expose through a **System.ComponentModel.TypeDescriptor** object.

If you are overriding this method you should call the base implementation before you perform your own filtering. When overriding this method, it should return the augmented set of attributes. If the method does not modify any attributes, it may just return a reference to its input parameter. If you do make a change to the attributes, you must create a new array. The set of events for the component.



## PreFilterProperties

[C#] protected override void PreFilterProperties(IDictionary properties);  
[C++] protected: void PreFilterProperties(IDictionary\* properties);  
[VB] Overrides Protected Sub PreFilterProperties(ByVal properties As  
IDictionary)  
[JScript] protected override function PreFilterProperties(properties : IDictionary);

### *Description*

Allows a designer to filter the set of member attributes that the component it is designing will expose through a **System.ComponentModel.TypeDescriptor** object.

If you are overriding this method you should call the base implementation before you perform your own filtering. When overriding this method, it should return the augmented set of attributes. If the method does not modify any attributes, it may just return a reference to its input parameter. If you do make a change to the attributes, you must create a new array. The set of properties to filter for the component.

HtmlIntrinsicControlDesigner class (System.Web.UI.Design)

ToString

### *Description*

Provides a base designer for all intrinsic HTML controls.

1        Intrinsic HTML controls are defined within, or derived from classes  
2 defined within, the **System.Web.UI.HtmlControls.HtmlControl** namespace.

3        **HtmlIntrinsicControlDesigner**

4        *Example Syntax:*

5        **ToString**

6  
7        [C#] **public HtmlIntrinsicControlDesigner();**

8        [C++] **public: HtmlIntrinsicControlDesigner();**

9        [VB] **Public Sub New()**

10        [JScript] **public function HtmlIntrinsicControlDesigner();**

11        **AssociatedComponents**

12        **Behavior**

13        **Component**

14        **DataBindings**

15        **DesignTimeElement**

16        **InheritanceAttribute**

17        **Inherited**

18        **ShadowProperties**

19        **ShouldCodeSerialize**

20        **Verbs**

21        **HyperLinkDataBindingHandler** class (**System.Web.UI.Design**)

22        **ToString**

23  
24  
25        *Description*

Provides a data binding handler for a hyperlink property.

HyperLinkDataBindingHandler

*Example Syntax:*

ToString

[C#] public HyperLinkDataBindingHandler();

[C++] public: HyperLinkDataBindingHandler();

[VB] Public Sub New()

[JScript] public function HyperLinkDataBindingHandler();

DataBindControl

[C#] public override void DataBindControl(IDesignerHost designerHost, Control control);

[C++] public: void DataBindControl(IDesignerHost\* designerHost, Control\* control);

[VB] Overrides Public Sub DataBindControl(ByVal designerHost As IDesignerHost, ByVal control As Control)

[JScript] public override function DataBindControl(designerHost : IDesignerHost, control : Control);

### *Description*

Adds this data binding to the specified control. The designer host for the document that contains the control. The control to add this data binding to.

IControlDesignerBehavior interface (System.Web.UI.Design)

ToString

1  
2  
3 *Description*

4 Provides an interface to provide a behavior for a control designer.

5 DesignTimeElementView

6 ToString

7  
8 [C#] object DesignTimeElementView {get;}

9 [C++] Object\* get\_DesignTimeElementView();

10 [VB] ReadOnly Property DesignTimeElementView As Object

11 [JScript] abstract function get DesignTimeElementView() : Object;

12  
13 *Description*

14 Gets or sets the design-time view control object for the designer.

15 DesignTimeHtml

16 ToString

17  
18 [C#] string DesignTimeHtml {get; set;}

19 [C++] String\* get\_DesignTimeHtml();void set\_DesignTimeHtml(String\*);

20 [VB] Property DesignTimeHtml As String

21 [JScript] abstract function get DesignTimeHtml() : String;public abstract function  
22 set DesignTimeHtml(String);

23  
24 *Description*

25 Gets or sets the design-time HTML for the designer's control.

## OnTemplateModeChanged

```
[C#] void OnTemplateModeChanged();  
[C++] void OnTemplateModeChanged();  
[VB] Sub OnTemplateModeChanged()  
[JScript] function OnTemplateModeChanged();
```

### *Description*

Raises the TemplateModeChanged event.

Raising an event invokes the event handler through a delegate. For more information, see .

IDataSourceProvider interface (System.Web.UI.Design)

## OnTemplateModeChanged

### *Description*

Provides an interface that enables access to a data source.

A class that can provide a data source can implement this interface to enable access to its data source by objects that use the **System.Web.UI.Design.IDataSourceProvider** interface.

## GetResolvedSelectedDataSource

```
[C#] IEnumerable GetResolvedSelectedDataSource();  
[C++] IEnumerable* GetResolvedSelectedDataSource();  
[VB] Function GetResolvedSelectedDataSource() As IEnumerable
```

1 [JScript] function GetResolvedSelectedDataSource() : IEnumerable;

3 *Description*

4 When implemented in a derived class, gets the selected data member from  
5 the selected data source.

6 *Return Value:* The selected data member from the selected data source.

7 GetSelectedDataSource

9 [C#] object GetSelectedDataSource();

10 [C++] Object\* GetSelectedDataSource();

11 [VB] Function GetSelectedDataSource() As Object

12 [JScript] function GetSelectedDataSource() : Object;

14 *Description*

15 When implemented in a derived class, gets a reference to the data source  
16 that this data source provider provides.

17 *Return Value:* The data source that this data source provider provides.

18 IHttpControlDesignerBehavior interface (System.Web.UI.Design)

19 GetSelectedDataSource

22 *Description*

23 Provides an interface to provide a behavior for an Interactive HTML  
24 control designer.

25 Designer

1           GetSelectedDataSource

2

3   [C#] HtmlControlDesigner Designer {get; set;}

4   [C++] HtmlControlDesigner\* get\_Designer();void

5   set\_Designer(HtmlControlDesigner\*);

6   [VB] Property Designer As HtmlControlDesigner

7   [JScript] abstract function get Designer() : HtmlControlDesigner;public abstract

8   function set Designer(HtmlControlDesigner);

9

10   *Description*

11           Gets or sets the designer that this behavior is associated with.

12           DesignTimeElement

13           GetSelectedDataSource

14

15   [C#] object DesignTimeElement {get;}

16   [C++] Object\* get\_DesignTimeElement();

17   [VB] ReadOnly Property DesignTimeElement As Object

18   [JScript] abstract function get DesignTimeElement() : Object;

19

20   *Description*

21           Gets the element that this designer is designing.

22           GetAttribute

23

24   [C#] object GetAttribute(string attribute, bool ignoreCase);

25   [C++] Object\* GetAttribute(String\* attribute, bool ignoreCase);

1 [VB] Function GetAttribute(ByVal attribute As String, ByVal ignoreCase As  
2 Boolean) As Object

3 [JScript] function GetAttribute(attribute : String, ignoreCase : Boolean) : Object;

4  
5 *Description*

6 Gets the specified attribute.

7 *Return Value:* The attribute that was retrieved. The attribute to retrieve. **true** if the  
8 attribute syntax is case-insensitive; otherwise, **false**.

9 GetStyleAttribute

10  
11 [C#] object GetStyleAttribute(string attribute, bool designTimeOnly, bool  
12 ignoreCase);

13 [C++] Object\* GetStyleAttribute(String\* attribute, bool designTimeOnly, bool  
14 ignoreCase);

15 [VB] Function GetStyleAttribute(ByVal attribute As String, ByVal  
16 designTimeOnly As Boolean, ByVal ignoreCase As Boolean) As Object

17 [JScript] function GetStyleAttribute(attribute : String, designTimeOnly : Boolean,  
18 ignoreCase : Boolean) : Object;

19  
20 *Description*

21 Gets the specified style attribute.

22 *Return Value:* The style attribute that was retrieved. The style attribute to retrieve.  
23 **true** if the attribute is a design-time only attribute; otherwise, **false**. **true** if the  
24 attribute syntax is case-insensitive; otherwise, **false**.

25 RemoveAttribute



```

1
2 [C#] void RemoveAttribute(string attribute, bool ignoreCase);
3 [C++] void RemoveAttribute(String* attribute, bool ignoreCase);
4 [VB] Sub RemoveAttribute(ByVal attribute As String, ByVal ignoreCase As
5 Boolean)
6 [JScript] function RemoveAttribute(attribute : String, ignoreCase : Boolean);
7

```

#### *Description*

Removes the specified attribute. The attribute to remove. **true** if the attribute syntax is case-insensitive; otherwise, **false**.

#### **RemoveStyleAttribute**

```

13 [C#] void RemoveStyleAttribute(string attribute, bool designTimeOnly, bool
14 ignoreCase);
15 [C++] void RemoveStyleAttribute(String* attribute, bool designTimeOnly, bool
16 ignoreCase);
17 [VB] Sub RemoveStyleAttribute(ByVal attribute As String, ByVal
18 designTimeOnly As Boolean, ByVal ignoreCase As Boolean)
19 [JScript] function RemoveStyleAttribute(attribute : String, designTimeOnly :
20 Boolean, ignoreCase : Boolean);
21

```

#### *Description*

Removes the specified style attribute. The style attribute to remove. **true** if the attribute is a design-time only attribute; otherwise, **false**. **true** if the attribute syntax is case-insensitive; otherwise, **false**.

1           SetAttribute

2

3   [C#] void SetAttribute(string attribute, object value, bool ignoreCase);

4   [C++] void SetAttribute(String\* attribute, Object\* value, bool ignoreCase);

5   [VB] Sub SetAttribute(ByVal attribute As String, ByVal value As Object, ByVal

6   ignoreCase As Boolean)

7   [JScript] function SetAttribute(attribute : String, value : Object, ignoreCase :

8   Boolean);

9

10   *Description*

11           Sets the specified attribute to the specified object. The attribute to set. The

12   object on which to set the attribute. **true** if the attribute syntax is case-insensitive;

13   otherwise, **false**.

14           SetStyleAttribute

15

16   [C#] void SetStyleAttribute(string attribute, bool designTimeOnly, object value,

17   bool ignoreCase);

18   [C++] void SetStyleAttribute(String\* attribute, bool designTimeOnly, Object\*

19   value, bool ignoreCase);

20   [VB] Sub SetStyleAttribute(ByVal attribute As String, ByVal designTimeOnly As

21   Boolean, ByVal value As Object, ByVal ignoreCase As Boolean)

22   [JScript] function SetStyleAttribute(attribute : String, designTimeOnly : Boolean,

23   value : Object, ignoreCase : Boolean);

24

25   *Description*

1       Sets the specified style attribute to the specified object. The attribute to set.  
2     **true** if the attribute is a design-time only attribute; otherwise, **false**. The object to  
3     set the attribute on. **true** if the attribute syntax is case-insensitive; otherwise, **false**.

4       ImageUrlEditor class (System.Web.UI.Design)

5       SetStyleAttribute

6  
7  
8     *Description*

9       Provides a user interface for selecting a URL.

10      This class extends **System.Web.UI.Design.UrlEditor** and provides a  
11      caption and filter property.

12      ImageUrlEditor

13      *Example Syntax:*

14      SetStyleAttribute

15  
16     [C#] public ImageUrlEditor();

17     [C++] public: ImageUrlEditor();

18     [VB] Public Sub New()

19     [JavaScript] public function ImageUrlEditor();

20       Caption

21       SetStyleAttribute

22  
23     [C#] protected override string Caption {get;}

24     [C++] protected: \_\_property virtual String\* get\_Caption();

25     [VB] Overrides Protected ReadOnly Property Caption As String

1 [JScript] protected function get Caption() : String;

3 *Description*

4 Gets the caption for the editor.

5 Filter

6 SetStyleAttribute

8 [C#] protected override string Filter {get;}

9 [C++] protected: \_\_property virtual String\* get\_Filter();

10 [VB] Overrides Protected ReadOnly Property Filter As String

11 [JScript] protected function get Filter() : String;

13 *Description*

14 Gets the filter to use for filtering the file list.

15 Options

16 ITemplateEditingFrame interface (System.Web.UI.Design)

17 ToString

20 *Description*

21 Provides an interface to manage a template editing area.

22 ControlStyle

23 ToString

25 [C#] Style ControlStyle {get;}

1 [C++] Style\* get\_ControlStyle();  
2 [VB] ReadOnly Property ControlStyle As Style  
3 [JScript] abstract function get ControlStyle() : Style;  
4

5 *Description*

6 Gets or sets the style for the editing frame.

7 InitialHeight

8 ToString

9  
10 [C#] int InitialHeight {get; set;}

11 [C++] int get\_InitialHeight();void set\_InitialHeight(int);

12 [VB] Property InitialHeight As Integer

13 [JScript] abstract function get InitialHeight() : int;public abstract function set  
14 InitialHeight(int);

15  
16 *Description*

17 Gets or sets the initial height for the control.

18 InitialWidth

19 ToString

20  
21 [C#] int InitialWidth {get; set;}

22 [C++] int get\_InitialWidth();void set\_InitialWidth(int);

23 [VB] Property InitialWidth As Integer

24 [JScript] abstract function get InitialWidth() : int;public abstract function set  
25 InitialWidth(int);

1  
2 *Description*

3 Gets or sets the initial width for the control.

4 Name

5 ToString

6  
7 [C#] string Name {get;}

8 [C++] String\* get\_Name();

9 [VB] ReadOnly Property Name As String

10 [JScript] abstract function get Name() : String;

11  
12 *Description*

13 Gets or sets the name for this editing frame.

14 TemplateNames

15 ToString

16  
17 [C#] string[] TemplateNames {get;}

18 [C++] String\* get\_TemplateNames();

19 [VB] ReadOnly Property TemplateNames As String ()

20 [JScript] abstract function get TemplateNames() : String[];

21  
22 *Description*

23 Gets or sets a set of names of templates to use.

24 TemplateStyles

25 ToString

```

1
2 [C#] Style[] TemplateStyles {get;}
3 [C++] Style* get_TemplateStyles();
4 [VB] ReadOnly Property TemplateStyles As Style ()
5 [JScript] abstract function get TemplateStyles() : Style[];
6

```

### *Description*

Gets or sets the template styles for this control.

Verb

ToString

```

11
12 [C#] TemplateEditingVerb Verb {get; set;}
13 [C++] TemplateEditingVerb* get_Verb();void set_Verb(TemplateEditingVerb*);
14 [VB] Property Verb As TemplateEditingVerb
15 [JScript] abstract function get Verb() : TemplateEditingVerb;public abstract
16 function set Verb(TemplateEditingVerb);
17

```

### *Description*

Gets or sets the verb to invoke when editing the template.

Close

```

21
22 [C#] void Close(bool saveChanges);
23 [C++] void Close(bool saveChanges);
24 [VB] Sub Close(ByVal saveChanges As Boolean)
25 [JScript] function Close(saveChanges : Boolean);

```

1  
2 *Description*

3 Closes the control and saves any changes to the control if the specified  
4 value is **true** . **true** if any changes to the document should be changed; otherwise,  
5 **false**.

6 Open

7  
8 [C#] void Open();  
9 [C++] void Open();  
10 [VB] Sub Open()  
11 [JScript] function Open();  
12

13 *Description*

14 Opens and displays the control.

15 Resize

16  
17 [C#] void Resize(int width, int height);  
18 [C++] void Resize(int width, int height);  
19 [VB] Sub Resize(ByVal width As Integer, ByVal height As Integer)  
20 [JScript] function Resize(width : int, height : int);  
21

22 *Description*

23 Resizes the control to the specified width and height. The new width for the  
24 control. The new height for the control.

25 Save



```

1
2 [C#] void Save();
3 [C++] void Save();
4 [VB] Sub Save()
5 [JScript] function Save();
6

```

*Description*

Saves any changes to the document.

UpdateControlName

```

11 [C#] void UpdateControlName(string newName);
12 [C++] void UpdateControlName(String* newName);
13 [VB] Sub UpdateControlName(ByVal newName As String)
14 [JScript] function UpdateControlName(newName : String);
15

```

*Description*

Changes the name for the control to the specified name. The new name for the control.

ITemplateEditingService interface (System.Web.UI.Design)

UpdateControlName

*Description*

Provides services for editing templated control templates at design-time.

SupportsNestedTemplateEditing

## UpdateControlName

[C#] bool SupportsNestedTemplateEditing {get;}

[C++] bool get\_SupportsNestedTemplateEditing();

[VB] ReadOnly Property SupportsNestedTemplateEditing As Boolean

[JScript] abstract function get SupportsNestedTemplateEditing() : Boolean;

### *Description*

Indicates whether the service supports nested template editing.

This property indicates whether a template editor can be invoked from a parent template editor for a component of the parent template editor's templated control. This is called nested template editing, because a template editor is launched for a component of a control for which a template editor is already operating.

### CreateFrame

[C#] ITemplateEditingFrame CreateFrame(TemplatedControlDesigner designer, string frameName, string[] templateNames);

[C++] ITemplateEditingFrame\* CreateFrame(TemplatedControlDesigner\* designer, String\* frameName, String\* templateNames \_\_gc[]);

[VB] Function CreateFrame(ByVal designer As TemplatedControlDesigner, ByVal frameName As String, ByVal templateNames() As String) As

ITemplateEditingFrame

[JScript] function CreateFrame(designer : TemplatedControlDesigner, frameName : String, templateNames : String[]) : ITemplateEditingFrame; Creates a new

1 template editing frame.

2  
3 *Description*

4       Creates a new template editing frame for the specified  
5 **System.Web.UI.Design.TemplatedControlDesigner** , using the specified name  
6 and templates.

7 *Return Value:* The new **System.Web.UI.Design.ITemplateEditingFrame** . The  
8 designer that will use the template editing frame. The name of the editing frame  
9 that will be displayed on the frame. Typically this is the same as the  
10 **System.ComponentModel.Design.DesignerVerb.Text** property used as the  
11 menu text for the **System.Web.UI.Design.TemplateEditingVerb** that is invoked  
12 to create the frame. The names of the templates that the template editing frame  
13 will contain.

14       CreateFrame

15  
16 [C#] ITemplateEditingFrame CreateFrame(TemplatedControlDesigner designer,  
17 string frameName, string[] templateNames, Style controlStyle, Style[]  
18 templateStyles);

19 [C++] ITemplateEditingFrame\* CreateFrame(TemplatedControlDesigner\*  
20 designer, String\* frameName, String\* templateNames \_\_gc[], Style\* controlStyle,  
21 Style\* templateStyles[]);

22 [VB] Function CreateFrame(ByVal designer As TemplatedControlDesigner,  
23 ByVal frameName As String, ByVal templateNames() As String, ByVal  
24 controlStyle As Style, ByVal templateStyles() As Style) As  
25 ITemplateEditingFrame

[JScript] function CreateFrame(designer : TemplatedControlDesigner, frameName  
: String, templateNames : String[], controlStyle : Style, templateStyles : Style[]) :  
ITemplateEditingFrame;

#### *Description*

Creates a new template editing frame for the specified  
**System.Web.UI.Design.TemplatedControlDesigner** , using the specified name,  
template names, control style, and template styles.

*Return Value:* The new **System.Web.UI.Design.ITemplateEditingFrame** . The  
designer that will use the template editing frame. The name of the editing frame  
that will be displayed on the frame. Typically this is the same as the  
**System.ComponentModel.Design.DesignerVerb.Text** property used as the  
menu text for the **System.Web.UI.Design.TemplateEditingVerb** that is invoked  
to create the frame. The names of the templates that the template editing frame  
will contain. The control style for the editing frame. The template styles for the  
editing frame.

#### **GetContainingTemplateName**

[C#] string GetContainingTemplateName(Control control);  
[C++] String\* GetContainingTemplateName(Control\* control);  
[VB] Function GetContainingTemplateName(ByVal control As Control) As  
String  
[JScript] function GetContainingTemplateName(control : Control) : String;

#### *Description*

1 Gets the name of the parent template.  
2 *Return Value:* The name of the parent template. The control to get the name of the  
3 parent template of.

4 IWebFormReferenceManager interface (System.Web.UI.Design)

5 GetContainingTemplateName

6  
7  
8 *Description*

9 Provides an interface that can be used to manage references stored by a  
10 Web Form.

11 GetObjectType

12  
13 [C#] Type GetObjectType(string tagPrefix, string typeName);

14 [C++] Type\* GetObjectType(String\* tagPrefix, String\* typeName);

15 [VB] Function GetObjectType(ByVal tagPrefix As String, ByVal typeName As  
16 String) As Type

17 [JScript] function GetObjectType(tagPrefix : String, typeName : String) : Type;

18  
19 *Description*

20 Gets the type of the specified object.

21 *Return Value:* The **System.Type** of the object, if it could be resolved. The tag  
22 prefix for the type. The name of the type.

23 GetRegisterDirectives

24  
25 [C#] string GetRegisterDirectives();

1 [C++] String\* GetRegisterDirectives();

2 [VB] Function GetRegisterDirectives() As String

3 [JScript] function GetRegisterDirectives() : String;

4  
5 *Description*

6 Gets the register directives.

7 *Return Value:* The register directives for the project.

8 GetTagPrefix

9  
10 [C#] string GetTagPrefix(Type objectType);

11 [C++] String\* GetTagPrefix(Type\* objectType);

12 [VB] Function GetTagPrefix(ByVal objectType As Type) As String

13 [JScript] function GetTagPrefix(objectType : Type) : String;

14  
15 *Description*

16 Gets the tag prefix.

17 *Return Value:* The tag prefix for the specified object type, if it could be located.

18 The type of the object.

19 IWebFormsBuilderUIService interface (System.Web.UI.Design)

20 GetTagPrefix

21  
22  
23 *Description*

24 Provides methods to launch specific user interfaces for building properties  
25 at design-time.

## BuildColor

[C#] string BuildColor(Control owner, string initialColor);

[C++] String\* BuildColor(Control\* owner, String\* initialColor);

[VB] Function BuildColor(ByVal owner As Control, ByVal initialColor As String) As String

[JScript] function BuildColor(owner : Control, initialColor : String) : String;

### *Description*

Launches an editor to build a color property.

*Return Value:* A string that represents the color that was selected. The control that contains the property to build. The initial color for the editor to pre-select.

## BuildUrl

[C#] string BuildUrl(Control owner, string initialUrl, string baseUrl, string caption, string filter, UrlBuilderOptions options);

[C++] String\* BuildUrl(Control\* owner, String\* initialUrl, String\* baseUrl, String\* caption, String\* filter, UrlBuilderOptions options);

[VB] Function BuildUrl(ByVal owner As Control, ByVal initialUrl As String, ByVal baseUrl As String, ByVal caption As String, ByVal filter As String, ByVal options As UrlBuilderOptions) As String

[JScript] function BuildUrl(owner : Control, initialUrl : String, baseUrl : String, caption : String, filter : String, options : UrlBuilderOptions) : String;

### *Description*

1 Launches an editor to build a URL property.

2 *Return Value:* A string that contains the URL returned by the

3 **System.Web.UI.Design.UrlBuilder** . The control that contains the property to

4 build. The initial URL to display in the selection interface. The base URL to

5 display in the selection interface. A caption that presents a message in the

6 selection interface. A filter for the types of files listed in the selection interface.

7 The options for the **System.Web.UI.Design.UrlBuilder** to use.

8 IWebFormsDocumentService interface (System.Web.UI.Design)

9 BuildUrl

10  
11  
12 *Description*

13 Provides methods to access services for tracking the loading state of a web  
14 forms document, handling events at load time, accessing a document's location,  
15 managing a document's undo service, and setting a new selection within the  
16 document.

17 DocumentUrl

18 BuildUrl

19  
20 [C#] string DocumentUrl {get;}

21 [C++] String\* get\_DocumentUrl();

22 [VB] ReadOnly Property DocumentUrl As String

23 [JScript] abstract function get DocumentUrl() : String;

24  
25 *Description*



1 Gets or sets the URL at which the document is located.

2 IsLoading

3 BuildUrl

4  
5 [C#] bool IsLoading {get;}

6 [C++] bool get\_IsLoading();

7 [VB] ReadOnly Property IsLoading As Boolean

8 [JScript] abstract function get IsLoading() : Boolean;

9  
10 *Description*

11 Indicates whether the document service is currently loading.

12 BuildUrl

13  
14 [C#] event EventHandler LoadComplete;

15 [C++] \_\_event EventHandler\* LoadComplete;

16 [VB] Event LoadComplete As EventHandler

17  
18 *Description*

19 Occurs when the service has finished loading.

20 This event provides an opportunity to perform operations immediately after  
21 loading has completed. Events that should occur at load time can be registered as  
22 event handlers for this event.

23 CreateDiscardableUndoUnit

24  
25 [C#] object CreateDiscardableUndoUnit();

1 [C++] Object\* CreateDiscardableUndoUnit();  
2 [VB] Function CreateDiscardableUndoUnit() As Object  
3 [JScript] function CreateDiscardableUndoUnit() : Object;

4  
5 *Description*

6 Creates a discardable undo unit.

7 *Return Value:* The new discardable undo unit.

8 DiscardUndoUnit

9  
10 [C#] void DiscardUndoUnit(object discardableUndoUnit);  
11 [C++] void DiscardUndoUnit(Object\* discardableUndoUnit);  
12 [VB] Sub DiscardUndoUnit(ByVal discardableUndoUnit As Object)  
13 [JScript] function DiscardUndoUnit(discardableUndoUnit : Object);

14  
15 *Description*

16 Discards the specified undo unit. The undo unit to discard.

17 EnableUndo

18  
19 [C#] void EnableUndo(bool enable);  
20 [C++] void EnableUndo(bool enable);  
21 [VB] Sub EnableUndo(ByVal enable As Boolean)  
22 [JScript] function EnableUndo(enable : Boolean);

23  
24 *Description*  
25

1 Enables the ability to undo actions that occur within undoable action units  
2 or transactions. **true** if actions should be undoable; otherwise, **false**.

### 3 UpdateSelection

4  
5 [C#] void UpdateSelection();

6 [C++] void UpdateSelection();

7 [VB] Sub UpdateSelection()

8 [JScript] function UpdateSelection();

### 9 10 *Description*

11 When implemented in a derived class, updates the current selection.

12 When implemented in a derived class, this method updates the current  
13 selection.

14 ReadWriteControlDesigner class (System.Web.UI.Design)

### 15 UpdateSelection

### 16 17 18 *Description*

19 Provides design-time functionality for read/write server controls.

20 ReadWriteControlDesigner

### 21 *Example Syntax:*

### 22 UpdateSelection

23  
24 [C#] public ReadWriteControlDesigner();

25 [C++] public: ReadWriteControlDesigner();

1 [VB] Public Sub New()

2 [JScript] public function ReadWriteControlDesigner();

3  
4 *Description*

5       Initializes an instance of the  
6 **System.Web.UI.Design.ReadWriteControlDesigner** class.

7       AllowResize

8       AssociatedComponents

9       Behavior

10      Component

11      DataBindings

12      DesignTimeElement

13      DesignTimeElementView

14      DesignTimeHtmlRequiresLoadComplete

15      ID

16      InheritanceAttribute

17      Inherited

18      IsDirty

19      ReadOnly

20      ShadowProperties

21      ShouldCodeSerialize

22      Verbs

23      MapPropertyToStyle

24  
25 [C#] protected virtual void MapPropertyToStyle(string propName, object

1 varPropValue);

2 [C++] protected: virtual void MapPropertyToStyle(String\* propName, Object\*

3 varPropValue);

4 [VB] Overridable Protected Sub MapPropertyToStyle(ByVal propName As

5 String, ByVal varPropValue As Object)

6 [JScript] protected function MapPropertyToStyle(propName : String,

7 varPropValue : Object);

8  
9 *Description*

10 Maps a property, including description and value, to an Interactive HTML  
11 style.

12 *Return Value:* This method must be overridden in a derived class to implement the  
13 designer. The name of the property to map. The value of the property.

14 OnBehaviorAttached

15  
16 [C#] protected override void OnBehaviorAttached();

17 [C++] protected: void OnBehaviorAttached();

18 [VB] Overrides Protected Sub OnBehaviorAttached()

19 [JScript] protected override function OnBehaviorAttached();

20  
21 *Description*

22 Provides notification that is raised upon a behavior being attached to the  
23 designer.

24 OnComponentChanged

25

```

1
2 [C#] public override void OnComponentChanged(object sender,
3 ComponentChangedEventArgs ce);
4 [C++] public: void OnComponentChanged(Object* sender,
5 ComponentChangedEventArgs* ce);
6 [VB] Overrides Public Sub OnComponentChanged(ByVal sender As Object,
7 ByVal ce As ComponentChangedEventArgs)
8 [JScript] public override function OnComponentChanged(sender : Object, ce :
9 ComponentChangedEventArgs);
10

```

#### *Description*

Represents the method that will handle the **System.ComponentModel.Design.IComponentChangeService.ComponentChanged** event of the **System.ComponentModel.Design.IComponentChangeService**.

The **System.ComponentModel.Design.IComponentChangeService.ComponentChanged** event occurs after a property has been changed. This delegate allows implementors to do any post-processing that may be needed after a property change. The object sending the event. The event object used when raising a **System.ComponentModel.Design.IComponentChangeService.ComponentChanged** event notification.

TemplatedControlDesigner class (System.Web.UI.Design)

UpdateDesignTimeHtml

1  
2  
3 *Description*

4 Provides design-time functionality for template-based server controls.

5 TemplatedControlDesigner

6 *Example Syntax:*

7 UpdateDesignTimeHtml

8  
9 [C#] public TemplatedControlDesigner();

10 [C++] public: TemplatedControlDesigner();

11 [VB] Public Sub New()

12 [JScript] public function TemplatedControlDesigner();

13  
14 *Description*

15 Initializes a new instance of the

16 **System.Web.UI.Design.TemplatedControlDesigner** class.

17 ActiveTemplateEditingFrame

18 UpdateDesignTimeHtml

19  
20 [C#] public ITemplateEditingFrame ActiveTemplateEditingFrame {get;}

21 [C++] public: \_\_property ITemplateEditingFrame\*

22 get\_ActiveTemplateEditingFrame();

23 [VB] Public ReadOnly Property ActiveTemplateEditingFrame As

24 ITemplateEditingFrame

25 [JScript] public function get ActiveTemplateEditingFrame() :

ITemplateEditingFrame;

*Description*

Gets the active template editing frame.

AllowResize

AssociatedComponents

Behavior

CanEnterTemplateMode

UpdateDesignTimeHtml

*Description*

Indicates whether or not this designer will allow the viewing or editing of templates. This property is read-only.

Component

DataBindings

DesignTimeElement

DesignTimeElementView

DesignTimeHtmlRequiresLoadComplete

HidePropertiesInTemplateMode

UpdateDesignTimeHtml

*Description*



1 Indicates whether the properties of the control will be hidden when the  
2 control is placed into template editing mode.

3 The **System.Web.UI.Design.ControlDesigner.ID** property is never  
4 hidden.

5 ID

6 InheritanceAttribute

7 Inherited

8 InTemplateMode

9 UpdateDesignTimeHtml

10  
11  
12 *Description*

13 Indicates whether or not the designer document is in template mode.

14 A document is in template mode when a template is currently being viewed  
15 or edited in the forms designer.

16 IsDirty

17 ReadOnly

18 ShadowProperties

19 ShouldCodeSerialize

20 Verbs

21 CreateTemplateEditingFrame

22  
23 [C#] protected abstract ITemplateEditingFrame

24 CreateTemplateEditingFrame(TemplateEditingVerb verb);

25 [C++] protected: virtual ITemplateEditingFrame\*

```

1 CreateTemplateEditingFrame(TemplateEditingVerb* verb) = 0;
2 [VB] MustOverride Protected Function CreateTemplateEditingFrame(ByVal verb
3 As TemplateEditingVerb) As ITemplateEditingFrame
4 [JScript] protected abstract function CreateTemplateEditingFrame(verb :
5 TemplateEditingVerb) : ITemplateEditingFrame;
6

```

### *Description*

Creates a template editing frame for the specified verb.

*Return Value:* The new template editing frame. The template editing verb to create a template editing frame for.

### *Dispose*

```

13 [C#] protected override void Dispose(bool disposing);
14 [C++] protected: void Dispose(bool disposing);
15 [VB] Overrides Protected Sub Dispose(ByVal disposing As Boolean)
16 [JScript] protected override function Dispose(disposing : Boolean);
17

```

### *Description*

Releases the unmanaged resources used by the **System.Web.UI.Design.TemplatedControlDesigner** and optionally releases the managed resources.

This method is called by the public **Dispose()** method and the **System.Object.Finalize** method. **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

### *EnterTemplateMode*

```

1
2 [C#] public void EnterTemplateMode(ITemplateEditingFrame
3 newTemplateEditingFrame);
4 [C++] public: void EnterTemplateMode(ITemplateEditingFrame*
5 newTemplateEditingFrame);
6 [VB] Public Sub EnterTemplateMode(ByVal newTemplateEditingFrame As
7 ITemplateEditingFrame)
8 [JScript] public function EnterTemplateMode(newTemplateEditingFrame :
9 ITemplateEditingFrame);
10

```

#### *Description*

Opens a particular template frame object for editing in the designer.

If already in template mode and editing a different template frame, this method calls

**System.Web.UI.Design.TemplatedControlDesigner.ExitTemplateMode(System.Boolean, System.Boolean, System.Boolean)** to save all its templates and close that frame. The template editing frame object to open in the designer.

#### **ExitTemplateMode**

```

19
20 [C#] public void ExitTemplateMode(bool fSwitchingTemplates, bool fNested,
21 bool fSave);
22 [C++] public: void ExitTemplateMode(bool fSwitchingTemplates, bool fNested,
23 bool fSave);
24 [VB] Public Sub ExitTemplateMode(ByVal fSwitchingTemplates As Boolean,
25 ByVal fNested As Boolean, ByVal fSave As Boolean)

```

1 [JScript] public function ExitTemplateMode(fSwitchingTemplates : Boolean,  
2 fNested : Boolean, fSave : Boolean);

3  
4 *Description*

5 Closes the currently active template editing frame after saving any relevant  
6 changes. **true** when switching from one template editing frame to another,  
7 otherwise **false**. **true** if this designer is nested (one or more levels) within another  
8 control whose designer is also in template editing mode; otherwise **false**. **true** if  
9 templates should be saved on exit; otherwise, **false**.

10 *GetCachedTemplateEditingVerbs*

11  
12 [C#] protected abstract TemplateEditingVerb[]

13 GetCachedTemplateEditingVerbs();

14 [C++] protected: virtual TemplateEditingVerb\*

15 GetCachedTemplateEditingVerbs() [] = 0;

16 [VB] MustOverride Protected Function GetCachedTemplateEditingVerbs() As  
17 TemplateEditingVerb()

18 [JScript] protected abstract function GetCachedTemplateEditingVerbs() :  
19 TemplateEditingVerb[];

20  
21 *Description*

22 Gets the cached template editing verbs.

23 *Return Value:* An array of **System.Web.UI.Design.TemplateEditingVerb**  
24 objects, if any.

25 *GetPersistInnerHtml*

1  
2 [C#] public override string GetPersistInnerHtml();

3 [C++] public: String\* GetPersistInnerHtml();

4 [VB] Overrides Public Function GetPersistInnerHtml() As String

5 [JScript] public override function GetPersistInnerHtml() : String;

6  
7 *Description*

8 Gets the HTML to be persisted for the content present within the associated  
9 server control runtime.

10 *Return Value:* The persistable inner HTML.

11 GetTemplateContainerDataItemProperty

12  
13 [C#] public virtual string GetTemplateContainerDataItemProperty(string  
14 templateName);

15 [C++] public: virtual String\* GetTemplateContainerDataItemProperty(String\*  
16 templateName);

17 [VB] Overridable Public Function GetTemplateContainerDataItemProperty(ByVal  
18 templateName As String) As String

19 [JScript] public function GetTemplateContainerDataItemProperty(templateName :  
20 String) : String;

21  
22 *Description*

23 Gets the data item property of the template's container.

24 *Return Value:* A string representing the data. The name of the template.

25 GetTemplateContainerDataSource

```

1 [C#] public virtual IEnumerable GetTemplateContainerDataSource(string
2 templateName);
3
4 [C++] public: virtual IEnumerable* GetTemplateContainerDataSource(String*
5 templateName);
6
7 [VB] Overridable Public Function GetTemplateContainerDataSource(ByVal
8 templateName As String) As IEnumerable
9
10 [JScript] public function GetTemplateContainerDataSource(templateName :
11 String) : IEnumerable;
12

```

### *Description*

Gets the data source of the template's container.

*Return Value:* The data source of the container of the specified template. The name of the template.

### *GetTemplateContent*

```

17 [C#] public abstract string GetTemplateContent(ITemplateEditingFrame
18 editingFrame, string templateName, out bool allowEditing);
19
20 [C++] public: virtual String* GetTemplateContent(ITemplateEditingFrame*
21 editingFrame, String* templateName, bool* allowEditing) = 0;
22
23 [VB] MustOverride Public Function GetTemplateContent(ByVal editingFrame As
24 ITemplateEditingFrame, ByVal templateName As String, ByRef allowEditing As
25 Boolean) As String
26
27 [JScript] public abstract function GetTemplateContent(editingFrame :
28 ITemplateEditingFrame, templateName : String, allowEditing : Boolean) : String;
29

```

1  
2 *Description*

3 Gets the template's content.

4 *Return Value:* The content of the template. The template editing frame to retrieve  
5 the content of. The name of the template. A boolean variable that will be set to  
6 **true** if the template's content can be edited, or **false** if the content is read-only.

7 *GetTemplateEditingVerbs*

8  
9 [C#] public TemplateEditingVerb[] GetTemplateEditingVerbs();  
10 [C++] public: TemplateEditingVerb\* GetTemplateEditingVerbs() [];  
11 [VB] Public Function GetTemplateEditingVerbs() As TemplateEditingVerb()  
12 [JScript] public function GetTemplateEditingVerbs() : TemplateEditingVerb[];

13  
14 *Description*

15 Gets the template editing verbs available to the designer.

16 *Return Value:* The template editing verbs available to the designer.

17 *GetTemplateFromText*

18  
19 [C#] protected ITemplate GetTemplateFromText(string text);  
20 [C++] protected: ITemplate\* GetTemplateFromText(String\* text);  
21 [VB] Protected Function GetTemplateFromText(ByVal text As String) As  
22 ITemplate  
23 [JScript] protected function GetTemplateFromText(text : String) : ITemplate;

24  
25 *Description*

Creates a template from the specified text.

*Return Value:* An **System.Web.UI.ITemplate** from the specified text. The text to retrieve a template from.

#### GetTemplatePropertyParentType

[C#] public virtual Type GetTemplatePropertyParentType(string templateName);

[C++] public: virtual Type\* GetTemplatePropertyParentType(String\* templateName);

[VB] Overridable Public Function GetTemplatePropertyParentType(ByVal templateName As String) As Type

[JScript] public function GetTemplatePropertyParentType(templateName : String) : Type;

#### Description

Gets the type of the parent of the template property.

*Return Value:* The type of the object that has the template property. The name of the template to return the type of the parent for.

#### GetTextFromTemplate

[C#] protected string GetTextFromTemplate(ITemplate template);

[C++] protected: String\* GetTextFromTemplate(ITemplate\* template);

[VB] Protected Function GetTextFromTemplate(ByVal template As ITemplate) As String

[JScript] protected function GetTextFromTemplate(template : ITemplate) : String;



1  
2 *Description*

3 Gets a string of text that represents the specified template.

4 *Return Value:* A string that represents the specified template. The

5 **System.Web.UI.ITemplate** to convert to text.

6 **OnBehaviorAttached**

7  
8 [C#] protected override void OnBehaviorAttached();

9 [C++] protected: void OnBehaviorAttached();

10 [VB] Overrides Protected Sub OnBehaviorAttached()

11 [JScript] protected override function OnBehaviorAttached();

12  
13 *Description*

14 Notification that is called when the behavior is attached to the designer.

15 **OnComponentChanged**

16  
17 [C#] public override void OnComponentChanged(object sender,

18 ComponentChangedEventArgs ce);

19 [C++] public: void OnComponentChanged(Object\* sender,

20 ComponentChangedEventArgs\* ce);

21 [VB] Overrides Public Sub OnComponentChanged(ByVal sender As Object,

22 ByVal ce As ComponentChangedEventArgs)

23 [JScript] public override function OnComponentChanged(sender : Object, ce :

24 ComponentChangedEventArgs);

1  
2 *Description*

3 Delegate to handle the component changed event.

4 This is called after a property has been changed. It allows the implementor  
5 to do any post-processing that may be needed after a property change. The object  
6 sending the event. A

7 **System.ComponentModel.Design.ComponentChangedEventArgs** that  
8 provides data for the event.

9 OnSetParent

10  
11 [C#] public override void OnSetParent();  
12 [C++] public: void OnSetParent();  
13 [VB] Overrides Public Sub OnSetParent()  
14 [JScript] public override function OnSetParent();  
15

16 *Description*

17 Notification that is called when the associated control is parented.

18 OnTemplateModeChanged

19  
20 [C#] protected virtual void OnTemplateModeChanged();  
21 [C++] protected: virtual void OnTemplateModeChanged();  
22 [VB] Overridable Protected Sub OnTemplateModeChanged()  
23 [JScript] protected function OnTemplateModeChanged();  
24

25 *Description*

Provides notification and handles processing that should occur when the template mode is changed.

### PreFilterProperties

[C#] protected override void PreFilterProperties(IDictionary properties);

[C++] protected: void PreFilterProperties(IDictionary\* properties);

[VB] Overrides Protected Sub PreFilterProperties(ByVal properties As IDictionary)

[JScript] protected override function PreFilterProperties(properties : IDictionary);

### *Description*

Allows a designer to filter the set of member attributes the component it is designing will expose through a **System.ComponentModel.TypeDescriptor** object.

*Return Value:* The augmented set of attributes. If the method does not modify any attributes, it may just return a reference to its input parameter. If you do make a change to the attributes, you must create a new array.

This method is called immediately before its corresponding "Post" method. If you are overriding this method you should call the base implementation before you perform your own filtering. The member attributes for component.

### SaveActiveTemplateEditingFrame

[C#] protected void SaveActiveTemplateEditingFrame();

[C++] protected: void SaveActiveTemplateEditingFrame();

[VB] Protected Sub SaveActiveTemplateEditingFrame()

1 [JScript] protected function SaveActiveTemplateEditingFrame();

3 *Description*

4 Saves the active template editing frame.

5 SetTemplateContent

7 [C#] public abstract void SetTemplateContent(ITemplateEditingFrame  
8 editingFrame, string templateName, string templateContent);

9 [C++] public: virtual void SetTemplateContent(ITemplateEditingFrame\*  
10 editingFrame, String\* templateName, String\* templateContent) = 0;

11 [VB] MustOverride Public Sub SetTemplateContent(ByVal editingFrame As  
12 ITemplateEditingFrame, ByVal templateName As String, ByVal templateContent  
13 As String)

14 [JScript] public abstract function SetTemplateContent(editingFrame :  
15 ITemplateEditingFrame, templateName : String, templateContent : String);

17 *Description*

18 Sets the specified template's content to the specified content. The template  
19 editing frame to provide content for. The name of the template. The content to set  
20 for the template.

21 UpdateDesignTimeHtml

23 [C#] public override void UpdateDesignTimeHtml();

24 [C++] public: void UpdateDesignTimeHtml();

25 [VB] Overrides Public Sub UpdateDesignTimeHtml()

1 [JScript] public override function UpdateDesignTimeHtml();

3 *Description*

4 Updates the design-time HTML.

5 This method should be called to refresh the design-time display if the  
6 template frames have been changed without calling methods that update the  
7 design-time HTML automatically.

8 TemplateEditingService class (System.Web.UI.Design)

9 UpdateDesignTimeHtml

10 TemplateEditingService

11 *Example Syntax:*

12 UpdateDesignTimeHtml

13 SupportsNestedTemplateEditing

14 UpdateDesignTimeHtml

15 CreateFrame

17 [C#] public ITemplateEditingFrame CreateFrame(TemplatedControlDesigner  
18 designer, string frameName, string[] templateNames);

19 [C++] public: \_\_sealed ITemplateEditingFrame\*

20 CreateFrame(TemplatedControlDesigner\* designer, String\* frameName, String\*  
21 templateNames \_\_gc[]);

22 [VB] NotOverridable Public Function CreateFrame(ByVal designer As  
23 TemplatedControlDesigner, ByVal frameName As String, ByVal  
24 templateNames() As String) As ITemplateEditingFrame

```

1 [JScript] public function CreateFrame(designer : TemplatedControlDesigner,
2   frameName : String, templateNames : String[]) : ITemplateEditingFrame;
3     CreateFrame
4
5 [C#] public ITemplateEditingFrame CreateFrame(TemplatedControlDesigner
6   designer, string frameName, string[] templateNames, Style controlStyle, Style[]
7   templateStyles);
8 [C++] public: __sealed ITemplateEditingFrame*
9   CreateFrame(TemplatedControlDesigner* designer, String* frameName, String*
10  templateNames __gc[], Style* controlStyle, Style* templateStyles[]);
11 [VB] NotOverridable Public Function CreateFrame(ByVal designer As
12  TemplatedControlDesigner, ByVal frameName As String, ByVal
13  templateNames() As String, ByVal controlStyle As Style, ByVal templateStyles()
14  As Style) As ITemplateEditingFrame
15 [JScript] public function CreateFrame(designer : TemplatedControlDesigner,
16   frameName : String, templateNames : String[], controlStyle : Style, templateStyles
17   : Style[]) : ITemplateEditingFrame;
18     Dispose
19
20 [C#] public void Dispose();
21 [C++] public: __sealed void Dispose();
22 [VB] NotOverridable Public Sub Dispose()
23 [JScript] public function Dispose();
24     Finalize
25

```

```

1
2 [C#] ~TemplateEditingService();
3 [C++] ~TemplateEditingService();
4 [VB] Overrides Protected Sub Finalize()
5 [JScript] protected override function Finalize();
6     GetContainingTemplateName
7
8 [C#] public string GetContainingTemplateName(Control control);
9 [C++] public: __sealed String* GetContainingTemplateName(Control* control);
10 [VB] NotOverridable Public Function GetContainingTemplateName(ByVal
11 control As Control) As String
12 [JScript] public function GetContainingTemplateName(control : Control) : String;
13     TemplateEditingVerb class (System.Web.UI.Design)
14     ToString
15
16
17 Description
18     Represents a designer verb that creates a template editing frame, and can
19 only be invoked by a template editor.
20     TemplateEditingVerb
21     Example Syntax:
22     ToString
23
24 [C#] public TemplateEditingVerb(string text, int index,
25 TemplatedControlDesigner designer);

```

```

1 [C++] public: TemplateEditingVerb(String* text, int index,
2   TemplatedControlDesigner* designer);
3 [VB] Public Sub New(ByVal text As String, ByVal index As Integer, ByVal
4   designer As TemplatedControlDesigner)
5 [JScript] public function TemplateEditingVerb(text : String, index : int, designer :
6   TemplatedControlDesigner);
7

```

### *Description*

Initializes a new instance of the **System.Web.UI.Design.TemplateEditingVerb** class. The text to show for this verb on a menu. An optional integer value that can be used by a designer; typically to indicate the index of the verb within a set of verbs. The designer that can use this verb.

Checked

CommandID

Enabled

Index

ToString

### *Description*

Gets or sets the index, or other user data, for this verb.

This property is sometimes used to store the index of each verb in a set of verbs. When implementing a custom designer, this property can be used to store integer data in a custom defined manner.



1 OleStatus  
2 Supported  
3 Text  
4 Visible  
5 Dispose

6  
7 [C#] public void Dispose();  
8 [C++] public: \_\_sealed void Dispose();  
9 [VB] NotOverridable Public Sub Dispose()  
10 [JScript] public function Dispose(); Releases all resources used by the  
11 **System.Web.UI.Design.TemplateEditingVerb** .  
12

13 *Description*

14 Releases all resources used by the  
15 **System.Web.UI.Design.TemplateEditingVerb** .

16 Calling **System.Web.UI.Design.TemplateEditingVerb.Dispose** allows  
17 the resources used by the **System.Web.UI.Design.TemplateEditingVerb** to be  
18 reallocated for other purposes. For more information about  
19 **System.Web.UI.Design.TemplateEditingVerb.Dispose** , see .

20 Dispose

21  
22 [C#] protected virtual void Dispose(bool disposing);  
23 [C++] protected: virtual void Dispose(bool disposing);  
24 [VB] Overridable Protected Sub Dispose(ByVal disposing As Boolean)  
25 [JScript] protected function Dispose(disposing : Boolean);

1  
2 *Description*

3 Releases the unmanaged resources used by the  
4 **System.Web.UI.Design.TemplateEditingVerb** and optionally releases the  
5 managed resources.

6 This method is called by the public **Dispose()** method and the  
7 **System.Object.Finalize** method. **true** to release both managed and unmanaged  
8 resources; **false** to release only unmanaged resources.

9 Finalize

10  
11 [C#] ~TemplateEditingVerb();

12 [C++] ~TemplateEditingVerb();

13 [VB] Overrides Protected Sub Finalize()

14 [JScript] protected override function Finalize();

15 TextControlDesigner class (System.Web.UI.Design)

16 ToString

17  
18  
19 *Description*

20 Provides design-time support for the ASP.NET  
21 **System.Web.UI.WebControls.Label** and **System.Windows.Forms.HyperLink**  
22 server controls.

23 TextControlDesigner

24 *Example Syntax:*

25 ToString



1 [VB] Overrides Public Function GetDesignTimeHtml() As String

2 [JScript] public override function GetDesignTimeHtml() : String;

3  
4 *Description*

5 Gets the HTML that is used to represent the associated control at design  
6 time.

7 *Return Value:* The HTML used to represent the control at design-time.

8 GetPersistInnerHtml

9  
10 [C#] public override string GetPersistInnerHtml();

11 [C++] public: String\* GetPersistInnerHtml();

12 [VB] Overrides Public Function GetPersistInnerHtml() As String

13 [JScript] public override function GetPersistInnerHtml() : String;

14  
15 *Description*

16 Gets the HTML to persist for the inner tags of the control.

17 *Return Value:* The HTML for the tags of the control.

18 Initialize

19  
20 [C#] public override void Initialize(IComponent component);

21 [C++] public: void Initialize(IComponent\* component);

22 [VB] Overrides Public Sub Initialize(ByVal component As IComponent)

23 [JScript] public override function Initialize(component : IComponent);

24  
25 *Description*

1        Initializes this designer with the specified component. The component for  
2 this designer to design.

3        TextDataBindingHandler class (System.Web.UI.Design)

4        UpdateDesignTimeHtml

7        *Description*

8        Provides a data binding handler for a text property of a control.

9        TextDataBindingHandler

10       *Example Syntax:*

11       UpdateDesignTimeHtml

13       [C#] public TextDataBindingHandler();

14       [C++] public: TextDataBindingHandler();

15       [VB] Public Sub New()

16       [JScript] public function TextDataBindingHandler();

17       DataBindControl

19       [C#] public override void DataBindControl(IDesignerHost designerHost, Control  
20 control);

21       [C++] public: void DataBindControl(IDesignerHost\* designerHost, Control\*  
22 control);

23       [VB] Overrides Public Sub DataBindControl(ByVal designerHost As  
24 IDesignerHost, ByVal control As Control)

25       [JScript] public override function DataBindControl(designerHost : IDesignerHost,

control : Control);

*Description*

Adds this data binding to the specified control. The designer host for the document that contains the control. The control to add this data binding to.

UrlBuilder class (System.Web.UI.Design)

ToString

*Description*

Launches a URL editor that allows a user to select a URL.

The

**System.Web.UI.Design.UrlBuilder.BuildUrl(System.ComponentModel.IComponent, System.Windows.Forms.Control, System.String, System.String, System.String)** method launches a user interface for selecting a URL.

BuildUrl

[C#] public static string BuildUrl(IComponent component, Control owner, string initialUrl, string caption, string filter);

[C++] public: static String\* BuildUrl(IComponent\* component, Control\* owner, String\* initialUrl, String\* caption, String\* filter);

[VB] Public Shared Function BuildUrl(ByVal component As IComponent, ByVal owner As Control, ByVal initialUrl As String, ByVal caption As String, ByVal filter As String) As String

[JScript] public static function BuildUrl(component : IComponent, owner :

Control, initialUrl : String, caption : String, filter : String) : String; Launches the  
Url Picker to build a Url property.

#### *Description*

Launches the Url Picker to build a Url property.

*Return Value:* The Url returned from the Url Picker. The component whose site is  
to be used to access design-time services. The control used to parent the picker  
window. The initial Url to be shown in the picker window. The caption of the  
picker window. The filter for selecting files in the picker window.

#### **BuildUrl**

[C#] public static string BuildUrl(IComponent component, Control owner, string  
initialUrl, string caption, string filter, UrlBuilderOptions options);

[C++] public: static String\* BuildUrl(IComponent\* component, Control\* owner,  
String\* initialUrl, String\* caption, String\* filter, UrlBuilderOptions options);

[VB] Public Shared Function BuildUrl(ByVal component As IComponent, ByVal  
owner As Control, ByVal initialUrl As String, ByVal caption As String, ByVal  
filter As String, ByVal options As UrlBuilderOptions) As String

[JScript] public static function BuildUrl(component : IComponent, owner :  
Control, initialUrl : String, caption : String, filter : String, options :  
UrlBuilderOptions) : String;

#### *Description*

Launches the Url Picker to build a Url property using the specified  
**System.Web.UI.Design.UrlBuilderOptions** .

*Return Value:* The Url returned from the Url Picker. The component whose site is to be used to access design-time services. The control used to parent the picker window. The initial Url to be shown in the picker window. The caption of the picker window. The filter for selecting files in the picker window. A **System.Web.UI.Design.UrlBuilderOptions** indicating the options for Url selection.

UrlBuilderOptions enumeration (System.Web.UI.Design)

ToString

#### *Description*

Defines identifiers for settings for a **System.Web.UI.Design.UrlBuilder**.

ToString

[C#] public const UrlBuilderOptions NoAbsolute;

[C++] public: const UrlBuilderOptions NoAbsolute;

[VB] Public Const NoAbsolute As UrlBuilderOptions

[JScript] public var NoAbsolute : UrlBuilderOptions;

#### *Description*

Build a URL that references a path relative to the current path, rather than one that consists of a fully qualified, absolute path reference URL.

ToString

[C#] public const UrlBuilderOptions None;



1 [C++] public: const UrlBuilderOptions None;

2 [VB] Public Const None As UrlBuilderOptions

3 [JScript] public var None : UrlBuilderOptions;

4  
5 *Description*

6 Use no additional options for the **System.Web.UI.Design.UrlBuilder** .

7 UrlEditor class (System.Web.UI.Design)

8 ToString

9  
10  
11 *Description*

12 Provides a user interface for selecting a URL.

13 UrlEditor

14 *Example Syntax:*

15 ToString

16  
17 [C#] public UrlEditor();

18 [C++] public: UrlEditor();

19 [VB] Public Sub New()

20 [JScript] public function UrlEditor();

21 Caption

22 ToString

23  
24 [C#] protected virtual string Caption {get;}

25 [C++] protected: \_\_property virtual String\* get\_Caption();

[VB] Overridable Protected ReadOnly Property Caption As String

[JScript] protected function get Caption() : String;

*Description*

Gets the caption for the URL.

Filter

ToString

[C#] protected virtual string Filter {get;}

[C++] protected: \_\_property virtual String\* get\_Filter();

[VB] Overridable Protected ReadOnly Property Filter As String

[JScript] protected function get Filter() : String;

*Description*

Gets or sets the file extensions to filter the file list for.

Options

ToString

[C#] protected virtual UrlBuilderOptions Options {get;}

[C++] protected: \_\_property virtual UrlBuilderOptions get\_Options();

[VB] Overridable Protected ReadOnly Property Options As UrlBuilderOptions

[JScript] protected function get Options() : UrlBuilderOptions;

*Description*

Gets the options for a URL builder to use.

## EditValue

```
[C#] public override object EditValue(ITypeDescriptorContext context,  
IServiceProvider provider, object value);  
[C++] public: Object* EditValue(ITypeDescriptorContext* context,  
IServiceProvider* provider, Object* value);  
[VB] Overrides Public Function EditValue(ByVal context As  
ITypeDescriptorContext, ByVal provider As IServiceProvider, ByVal value As  
Object) As Object  
[JScript] public override function EditValue(context : ITypeDescriptorContext,  
provider : IServiceProvider, value : Object) : Object;
```

### *Description*

Edits the value of the specified object using the editor style provided by the **System.Web.UI.Design.UrlEditor.GetEditStyle(System.ComponentModel.ITypeDescriptorContext)** method.

*Return Value:* The new value of the object. If the value of the object hasn't changed, this should return the same object it was passed.

A service provider is provided so that any required editing services can be obtained. An **System.ComponentModel.ITypeDescriptorContext** that can be used to gain additional context information. A service provider object through which editing services may be obtained. An instance of the value being edited.

## GetEditStyle

```
[C#] public override UITypeEditorEditStyle
```

```

1 GetEditStyle(ITypeDescriptorContext context);
2 [C++] public: UITypeEditorEditStyle GetEditStyle(ITypeDescriptorContext*
3 context);
4 [VB] Overrides Public Function GetEditStyle(ByVal context As
5 ITypeDescriptorContext) As UITypeEditorEditStyle
6 [JScript] public override function GetEditStyle(context : ITypeDescriptorContext)
7 : UITypeEditorEditStyle;
8

```

### *Description*

Gets the editing style of the **System.Web.UI.Design.UrlEditor.EditValue(System.ComponentModel.ITypeDescriptorContext, System.IServiceProvider, System.Object)** method.

*Return Value:* One of the **System.Drawing.Design.UITypeEditorEditStyle** values indicating the provided editing style. If the method is not supported, this will return **System.Drawing.Design.UITypeEditorEditStyle.None** . An **System.ComponentModel.ITypeDescriptorContext** that can be used to gain additional context information.

UserControlDesigner class (System.Web.UI.Design)

ToString

### *Description*

Provides design-time support for web user controls.

UserControlDesigner

*Example Syntax:*

1 ToString

2

3 [C#] public UserControlDesigner();

4 [C++] public: UserControlDesigner();

5 [VB] Public Sub New()

6 [JScript] public function UserControlDesigner();

7

8 *Description*

9 Initializes a new instance of the

10 **System.Web.UI.Design.UserControlDesigner** class.

11 AllowResize

12 ToString

13

14 [C#] public override bool AllowResize {get;}

15 [C++] public: \_\_property virtual bool get \_AllowResize();

16 [VB] Overrides Public ReadOnly Property AllowResize As Boolean

17 [JScript] public function get AllowResize() : Boolean;

18

19 *Description*

20 Indicates whether all user controls are resizable.

21 AssociatedComponents

22 Behavior

23 Component

24 DataBindings

25 DesignTimeElement

1 DesignTimeElementView  
2 DesignTimeHtmlRequiresLoadComplete  
3 ID  
4 InheritanceAttribute  
5 Inherited  
6 IsDirty  
7 ReadOnly  
8 ShadowProperties  
9 ShouldCodeSerialize  
10 ToString

11  
12  
13 *Description*

14 Indicates whether this designer should add a field declaration for the  
15 control in the code-behind file for the current design document.

16 Verbs

17 GetDesignTimeHtml

18  
19 [C#] public override string GetDesignTimeHtml();

20 [C++] public: String\* GetDesignTimeHtml();

21 [VB] Overrides Public Function GetDesignTimeHtml() As String

22 [JScript] public override function GetDesignTimeHtml() : String;

23  
24 *Description*  
25

Gets the HTML that is used to represent the control at design time.

*Return Value:* The HTML that is used to represent the control at design time.

GetPersistInnerHtml

[C#] public override string GetPersistInnerHtml();

[C++] public: String\* GetPersistInnerHtml();

[VB] Overrides Public Function GetPersistInnerHtml() As String

[JScript] public override function GetPersistInnerHtml() : String;

#### *Description*

Gets the inner HTML that is persisted for the control.

*Return Value:* The content to be persisted from within the control's tags.

WebControlToolboxItem class (System.Web.UI.Design)

UpdateDesignTimeHtml

#### *Description*

Provides a base class for a Web server control

**System.Drawing.Design.ToolboxItem .**

WebControlToolboxItem

*Example Syntax:*

UpdateDesignTimeHtml

[C#] public WebControlToolboxItem();

[C++] public: WebControlToolboxItem();

1 [VB] Public Sub New()

2 [JScript] public function WebControlToolboxItem(); Initializes a new instance of  
3 the **System.Web.UI.Design.WebControlToolboxItem** class.

4  
5 *Description*

6       Initializes a new instance of the  
7 **System.Web.UI.Design.WebControlToolboxItem** class.

8       The display name will be a short version of the type name.

9       WebControlToolboxItem

10 *Example Syntax:*

11       UpdateDesignTimeHtml

12  
13 [C#] public WebControlToolboxItem(Type type);

14 [C++] public: WebControlToolboxItem(Type\* type);

15 [VB] Public Sub New(ByVal type As Type)

16 [JScript] public function WebControlToolboxItem(type : Type);

17  
18 *Description*

19       Initializes a new instance of the  
20 **System.Web.UI.Design.WebControlToolboxItem** class. The fully qualified type  
21 name of the tool for this toolbox item

22       AssemblyName

23       Bitmap

24       DisplayName

25       Filter



```

1      Locked
2      TypeName
3      CreateComponentsCore
4
5      [C#] protected override IComponent[] CreateComponentsCore(IDesignerHost
6      host);
7      [C++] protected: IComponent* CreateComponentsCore(IDesignerHost* host) [];
8      [VB] Overrides Protected Function CreateComponentsCore(ByVal host As
9      IDesignerHost) As IComponent()
10     [JScript] protected override function CreateComponentsCore(host :
11     IDesignerHost) : IComponent[];
12

```

### *Description*

Creates objects from each Type contained in this **System.Drawing.Design.ToolboxItem** , and adds them to the specified designer.

*Return Value:* An array of created **System.ComponentModel.IComponent** objects.

You can override this method to alter the way components are created. The **System.ComponentModel.Design.IDesignerHost** to use to create the components.

### **Deserialize**

```

23     [C#] protected override void Deserialize(SerializationInfo info, StreamingContext
24     context);
25     [C++] protected: void Deserialize(SerializationInfo* info, StreamingContext

```

1 context);

2 [VB] Overrides Protected Sub Deserialize(ByVal info As SerializationInfo, ByVal  
3 context As StreamingContext)

4 [JScript] protected override function Deserialize(info : SerializationInfo, context :  
5 StreamingContext);

6  
7 *Description*

8  
9 **GetToolAttributeValue**

10  
11 [C#] public object GetToolAttributeValue(IDesignerHost host, Type  
12 attributeType);

13 [C++] public: Object\* GetToolAttributeValue(IDesignerHost\* host, Type\*  
14 attributeType);

15 [VB] Public Function GetToolAttributeValue(ByVal host As IDesignerHost,  
16 ByVal attributeType As Type) As Object

17 [JScript] public function GetToolAttributeValue(host : IDesignerHost,  
18 attributeType : Type) : Object;

19  
20 *Description*

21 Gets the value of the attribute of the specified type of the toolbox item.

22 *Return Value:* The value of the specified type of attribute.

23 The default implementation will throw an ArgumentException if the  
24 *attributeType* parameter is anything other than a PersistChildrenAttribute. The

designer host for the current design document. The type of attribute to retrieve the value of.

### GetToolHtml

[C#] public string GetToolHtml(IDesignerHost host);

[C++] public: String\* GetToolHtml(IDesignerHost\* host);

[VB] Public Function GetToolHtml(ByVal host As IDesignerHost) As String

[JScript] public function GetToolHtml(host : IDesignerHost) : String;

### *Description*

Gets the HTML associated with the tool.

*Return Value:* The HTML associated with the tool. The designer host for the current design document.

### GetToolType

[C#] public Type GetToolType(IDesignerHost host);

[C++] public: Type\* GetToolType(IDesignerHost\* host);

[VB] Public Function GetToolType(ByVal host As IDesignerHost) As Type

[JScript] public function GetToolType(host : IDesignerHost) : Type;

### *Description*

Gets the type of this toolbox item.

*Return Value:* The type of this toolbox item. The designer host for the current design document.

### Initialize

```

1
2 [C#] public override void Initialize(Type type);
3 [C++] public: void Initialize(Type* type);
4 [VB] Overrides Public Sub Initialize(ByVal type As Type)
5 [JScript] public override function Initialize(type : Type);
6

```

### *Description*

Initializes this toolbox item. The type of this Web server control toolbox item.

### Serialize

```

12 [C#] protected override void Serialize(SerializationInfo info, StreamingContext
13 context);
14 [C++] protected: void Serialize(SerializationInfo* info, StreamingContext
15 context);
16 [VB] Overrides Protected Sub Serialize(ByVal info As SerializationInfo, ByVal
17 context As StreamingContext)
18 [JScript] protected override function Serialize(info : SerializationInfo, context :
19 StreamingContext);
20

```

### *Description*

XmlFileEditor class (System.Web.UI.Design)  
ToString

1  
2  
3 *Description*

4 Provides a user interface for selecting an XML File.

5 XmlFileEditor

6 *Example Syntax:*

7 ToString

8  
9 [C#] public XmlFileEditor();

10 [C++] public: XmlFileEditor();

11 [VB] Public Sub New()

12 [JScript] public function XmlFileEditor();

13 EditValue

14  
15 [C#] public override object EditValue(ITypeDescriptorContext context,

16 IServiceProvider provider, object value);

17 [C++] public: Object\* EditValue(ITypeDescriptorContext\* context,

18 IServiceProvider\* provider, Object\* value);

19 [VB] Overrides Public Function EditValue(ByVal context As

20 ITypeDescriptorContext, ByVal provider As IServiceProvider, ByVal value As

21 Object) As Object

22 [JScript] public override function EditValue(context : ITypeDescriptorContext,

23 provider : IServiceProvider, value : Object) : Object;

24  
25 *Description*

Edits the value of the specified object using the specified service provider and context.

*Return Value:* The new value of the object. If the value of the object hasn't changed, this method should return the same object it was passed.

A service provider is provided so that any required editing services can be obtained. An **System.ComponentModel.ITypeDescriptorContext** that can be used to gain additional context information. A service provider object through which editing services may be obtained. An instance of the value being edited.

#### GetEditStyle

[C#] public override UITypeEditorEditStyle

GetEditStyle(ITypeDescriptorContext context);

[C++] public: UITypeEditorEditStyle GetEditStyle(ITypeDescriptorContext\* context);

[VB] Overrides Public Function GetEditStyle(ByVal context As

ITypeDescriptorContext) As UITypeEditorEditStyle

[JScript] public override function GetEditStyle(context : ITypeDescriptorContext) : UITypeEditorEditStyle;

#### Description

Gets the editor style used by the **System.Web.UI.Design.XmlFileEditor.EditValue(System.ComponentModel.ITypeDescriptorContext, System.IServiceProvider, System.Object)** method.

*Return Value:* An value indicating the provided editing style. An

1 **System.ComponentModel.ITypeDescriptorContext** that can be used to gain  
2 additional context information.

3 XmlUrlEditor class (System.Web.UI.Design)

4 ToString

7 *Description*

8 Provides a user interface for editing an URL embedded in XML.

9 XmlUrlEditor

10 *Example Syntax:*

11 ToString

13 [C#] public XmlUrlEditor();

14 [C++] public: XmlUrlEditor();

15 [VB] Public Sub New()

16 [JScript] public function XmlUrlEditor();

17 Caption

18 ToString

20 [C#] protected override string Caption {get;}

21 [C++] protected: \_\_property virtual String\* get\_Caption();

22 [VB] Overrides Protected ReadOnly Property Caption As String

23 [JScript] protected function get Caption() : String;

25 *Description*

1 Gets or sets the caption for the URL.  
2 Filter  
3 ToString  
4  
5 [C#] protected override string Filter {get;}  
6 [C++] protected: \_\_property virtual String\* get\_Filter();  
7 [VB] Overrides Protected ReadOnly Property Filter As String  
8 [JScript] protected function get Filter() : String;

9

10 *Description*

11 Gets or sets the file extensions to filter the file list for.

12 Options

13 ToString

14

15 [C#] protected override UrlBuilderOptions Options {get;}  
16 [C++] protected: \_\_property virtual UrlBuilderOptions get\_Options();  
17 [VB] Overrides Protected ReadOnly Property Options As UrlBuilderOptions  
18 [JScript] protected function get Options() : UrlBuilderOptions;

19

20 *Description*

21 Gets the options for a URL builder to use.

22 XslUrlEditor class (System.Web.UI.Design)

23 ToString  
24  
25



1  
2  
3 *Description*

4 Provides a user interface for selecting an XSL file.

5 XslUrlEditor

6 *Example Syntax:*

7 ToString

8  
9 [C#] public XslUrlEditor();

10 [C++] public: XslUrlEditor();

11 [VB] Public Sub New()

12 [JScript] public function XslUrlEditor();

13 Caption

14 ToString

15  
16 [C#] protected override string Caption {get;}

17 [C++] protected: \_\_property virtual String\* get\_Caption();

18 [VB] Overrides Protected ReadOnly Property Caption As String

19 [JScript] protected function get Caption() : String;

20  
21 *Description*

22 Gets or sets the caption for the URL.

23 Filter

24 ToString

[C#] protected override string Filter {get;}

[C++] protected: \_\_property virtual String\* get\_Filter();

[VB] Overrides Protected ReadOnly Property Filter As String

[JScript] protected function get Filter() : String;

#### *Description*

### **System.Web.UI.Design.WebControls**

#### *Description*

The **System.Web.UI.Design.WebControls** namespace contains classes that can be used to extend design-time support for Web server controls.

AdRotatorDesigner class (System.Web.UI.Design.WebControls)

#### *Description*

Provides design-time support for the **System.Web.UI.WebControls.AdRotator** Web server control.

Constructors:

AdRotatorDesigner

*Example Syntax:*

1  
2 [C#] public AdRotatorDesigner();  
3 [C++] public: AdRotatorDesigner();  
4 [VB] Public Sub New()  
5 [JScript] public function AdRotatorDesigner();  
6     Properties:  
7     AllowResize  
8     AssociatedComponents  
9     Behavior  
10    Component  
11    DataBindings  
12    DesignTimeElement  
13    DesignTimeElementView  
14    DesignTimeHtmlRequiresLoadComplete  
15    ID  
16    InheritanceAttribute  
17    Inherited  
18    IsDirty  
19    ReadOnly  
20    ShadowProperties  
21    ShouldCodeSerialize  
22    Verbs  
23    Methods:  
24    GetDesignTimeHtml  
25

```

1
2 [C#] public override string GetDesignTimeHtml();
3 [C++] public: String* GetDesignTimeHtml();
4 [VB] Overrides Public Function GetDesignTimeHtml() As String
5 [JScript] public override function GetDesignTimeHtml() : String;
6

```

#### *Description*

Gets the HTML that is used to represent the control at design time.

*Return Value:* The HTML that is used to represent the control at design time.

BaseDataListComponentEditor class

(System.Web.UI.Design.WebControls)

UpdateDesignTimeHtml

#### *Description*

Provides a base component editor class for the Web Forms

**System.Web.UI.WebControls.DataGrid** and

**System.Web.UI.WebControls.DataList** Web server controls.

BaseDataListComponentEditor

*Example Syntax:*

UpdateDesignTimeHtml

```

23 [C#] public BaseDataListComponentEditor(int initialPage);
24 [C++] public: BaseDataListComponentEditor(int initialPage);
25 [VB] Public Sub New(ByVal initialPage As Integer)

```

1 [JScript] public function BaseDataListComponentEditor(initialPage : int);

3 *Description*

4       Initializes a new instance of the  
5 **System.Web.UI.Design.WebControls.BaseDataListComponentEditor** class.  
6 The index of the initial page to display.

7       EditComponent

9 [C#] public override bool EditComponent(ITypeDescriptorContext context, object  
10 obj, IWin32Window parent);

11 [C++] public: bool EditComponent(ITypeDescriptorContext\* context, Object\*  
12 obj, IWin32Window\* parent);

13 [VB] Overrides Public Function EditComponent(ByVal context As  
14 ITypeDescriptorContext, ByVal obj As Object, ByVal parent As IWin32Window)  
15 As Boolean

16 [JScript] public override function EditComponent(context :  
17 ITypeDescriptorContext, obj : Object, parent : IWin32Window) : Boolean;

19 *Description*

20       Edits the specified component using the specified context descriptor and  
21 parent window.

22 *Return Value:* **true** if editing the component succeeded; otherwise, **false** . An  
23 **System.ComponentModel.ITypeDescriptorContext** that can be used to gain  
24 additional context information. The component to edit. The parent window.

25       GetInitialComponentEditorPageIndex

[C#] protected override int GetInitialComponentEditorPageIndex();

[C++] protected: int GetInitialComponentEditorPageIndex();

[VB] Overrides Protected Function GetInitialComponentEditorPageIndex() As Integer

[JScript] protected override function GetInitialComponentEditorPageIndex() : int;

#### *Description*

Gets the index of the initial component editor page.

*Return Value:* The index of the initial page.

BaseDataListDesigner class (System.Web.UI.Design.WebControls)

ToString

#### *Description*

Provides a base designer class for the

**System.Web.UI.WebControls.DataList** Web server control.

BaseDataListDesigner

*Example Syntax:*

ToString

[C#] public BaseDataListDesigner();

[C++] public: BaseDataListDesigner();

[VB] Public Sub New()

[JScript] public function BaseDataListDesigner();

1  
2 *Description*

3       Initializes a new instance of the  
4 **System.Web.UI.Design.WebControls.BaseDataListDesigner** class.

5       ActiveTemplateEditingFrame

6       AllowResize

7       AssociatedComponents

8       Behavior

9       CanEnterTemplateMode

10      Component

11      DataBindings

12      DataKeyField

13      ToString

14  
15  
16 *Description*

17       Gets or sets the value of the control's data key field.

18       The

19 **System.Web.UI.Design.WebControls.BaseDataListDesigner.DataKeyField**

20 property of this designer gets or sets the

21 **System.Web.UI.WebControls.DataList.DataKeyField** property of the

22 **System.Web.UI.WebControls.DataList** control that this designer is editing.

23       DataMember

24       ToString

1  
2 [C#] public string DataMember {get; set;}

3 [C++] public: \_\_property String\* get\_DataMember();public: \_\_property void

4 set\_DataMember(String\*);

5 [VB] Public Property DataMember As String

6 [JScript] public function get DataMember() : String;public function set

7 DataMember(String);

8  
9 *Description*

10 Gets or sets the value of the control's data member field.

11 The

12 **System.Web.UI.Design.WebControls.BaseDataListDesigner.DataMember**

13 property of this designer gets or sets the

14 **System.Web.UI.WebControls.DataList.DataMember** property of the

15 **System.Web.UI.WebControls.DataList** control that this designer is editing.

16 DataSource

17 ToString

18  
19 [C#] public string DataSource {get; set;}

20 [C++] public: \_\_property String\* get\_DataSource();public: \_\_property void

21 set\_DataSource(String\*);

22 [VB] Public Property DataSource As String

23 [JScript] public function get DataSource() : String;public function set

24 DataSource(String);



1  
2 *Description*

3 Gets or sets the value of the control's data source property.

4 The

5 **System.Web.UI.Design.WebControls.BaseDataListDesigner.DataSource**

6 property of this designer gets or sets the

7 **System.Web.UI.WebControls.DataList.DataSource** property of the

8 **System.Web.UI.WebControls.DataList** control that this designer is editing.

9 DesignTimeElement

10 DesignTimeElementView

11 DesignTimeHtmlRequiresLoadComplete

12 ToString

13  
14  
15 *Description*

16 Indicates whether loading must have completed to display the design-time  
17 HTML.

18 If this property is set to **true** , the control will not be displayed until the  
19 entire load has completed.

20 HidePropertiesInTemplateMode

21 ID

22 InheritanceAttribute

23 Inherited

24 InTemplateMode

25 IsDirty

ReadOnly

ShadowProperties

ShouldCodeSerialize

Verbs

ToString

*Description*

Gets or sets the collection of verbs available to this designer.

Dispose

[C#] protected override void Dispose(bool disposing);

[C++] protected: void Dispose(bool disposing);

[VB] Overrides Protected Sub Dispose(ByVal disposing As Boolean)

[JScript] protected override function Dispose(disposing : Boolean);

*Description*

Releases the unmanaged resources used by the **System.Web.UI.Design.WebControls.BaseDataListDesigner** and optionally releases the managed resources.

This method is called by the public **Dispose()** method and the **System.Object.Finalize** method. **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

GetDesignTimeDataSource



As IEnumerable, ByVal minimumRows As Integer, ByRef dummyDataSource As Boolean) As IEnumerable

[JScript] protected function GetDesignTimeDataSource(selectedDataSource : IEnumerable, minimumRows : int, dummyDataSource : Boolean) : IEnumerable;

#### *Description*

Gets a set of sample data that matches the schema of the selected data source.

*Return Value:* A live data source for use at design-time. The data source that will be used as a template for the format of the data. The minimum number of rows of sample data that the data source data should contain. A boolean variable that will be set to **true** if the returned data source contains dummy data, or **false** if the data source was populated with data from an actual data source.

#### *GetResolvedSelectedDataSource*

[C#] public IEnumerable GetResolvedSelectedDataSource();

[C++] public: \_\_sealed IEnumerable\* GetResolvedSelectedDataSource();

[VB] NotOverridable Public Function GetResolvedSelectedDataSource() As IEnumerable

[JScript] public function GetResolvedSelectedDataSource() : IEnumerable;

#### *Description*

Gets the data member that is currently selected within the data source currently bound to the control.

*Return Value:* The currently selected data member, or **null** if the control was not

bound to a data source, or the data source, site of the designer's component, or the container of the data source could not be accessed.

### GetSelectedDataSource

[C#] public object GetSelectedDataSource();

[C++] public: \_\_sealed Object\* GetSelectedDataSource();

[VB] NotOverridable Public Function GetSelectedDataSource() As Object

[JScript] public function GetSelectedDataSource() : Object;

### *Description*

Gets the selected data source component from the component's container.

*Return Value:* The selected data source, or **null** if a data source is not found, or if a data source with the same name does not exist.

### GetTemplateContainerDataSource

[C#] public override IEnumerable GetTemplateContainerDataSource(string  
templateName);

[C++] public: IEnumerable\* GetTemplateContainerDataSource(String\*  
templateName);

[VB] Overrides Public Function GetTemplateContainerDataSource(ByVal  
templateName As String) As IEnumerable

[JScript] public override function

GetTemplateContainerDataSource(templateName : String) : IEnumerable;

### *Description*

1 Gets the template's container's data source.

2 *Return Value:* An object that implements **System.Collections.IEnumerable** and

3 contains the data source or data sources available to the template's container. The

4 name of the template to retrieve the data source for.

#### 5 Initialize

6

7 [C#] public override void Initialize(IComponent component);

8 [C++] public: void Initialize(IComponent\* component);

9 [VB] Overrides Public Sub Initialize(ByVal component As IComponent)

10 [JScript] public override function Initialize(component : IComponent);

11

#### 12 Description

13 Initializes the designer with the **System.Web.UI.WebControls.DataGrid**

14 control that this instance of the designer is associated with. The associated

15 **System.Web.UI.WebControls.DataGrid** control.

#### 16 InvokePropertyBuilder

17

18 [C#] protected internal void InvokePropertyBuilder(int initialPage);

19 [C++] protected public: void InvokePropertyBuilder(int initialPage);

20 [VB] Protected Friend Dim Sub InvokePropertyBuilder(ByVal initialPage As

21 Integer)

22 [JScript] package function InvokePropertyBuilder(initialPage : int);

23

#### 24 Description

1 Invokes the property builder beginning with the specified page. The page to  
2 begin with.

### 3 OnAutoFormat

4  
5 [C#] protected void OnAutoFormat(object sender, EventArgs e);  
6 [C++] protected: void OnAutoFormat(Object\* sender, EventArgs\* e);  
7 [VB] Protected Sub OnAutoFormat(ByVal sender As Object, ByVal e As  
8 EventArgs)  
9 [JScript] protected function OnAutoFormat(sender : Object, e : EventArgs);  
10

#### 11 *Description*

12 Represents the method that will handle the AutoFormat event. The source  
13 of the event. An **System.EventArgs** object that provides data about the event.

### 14 OnComponentChanged

15  
16 [C#] public override void OnComponentChanged(object sender,  
17 ComponentChangedEventArgs e);  
18 [C++] public: void OnComponentChanged(Object\* sender,  
19 ComponentChangedEventArgs\* e);  
20 [VB] Overrides Public Sub OnComponentChanged(ByVal sender As Object,  
21 ByVal e As ComponentChangedEventArgs)  
22 [JScript] public override function OnComponentChanged(sender : Object, e :  
23 ComponentChangedEventArgs);  
24

#### 25 *Description*

Represents the method that will handle the component change event. The source of the event. The

**System.ComponentModel.Design.ComponentChangedEventArgs** that provides data about the event.

#### OnDataSourceChanged

[C#] protected internal virtual void OnDataSourceChanged();

[C++] protected public: virtual void OnDataSourceChanged();

[VB] Overridable Protected Friend Dim Sub OnDataSourceChanged()

[JScript] package function OnDataSourceChanged();

#### *Description*

Raises the DataSourceChanged event.

#### OnPropertyBuilder

[C#] protected void OnPropertyBuilder(object sender, EventArgs e);

[C++] protected: void OnPropertyBuilder(Object\* sender, EventArgs\* e);

[VB] Protected Sub OnPropertyBuilder(ByVal sender As Object, ByVal e As EventArgs)

[JScript] protected function OnPropertyBuilder(sender : Object, e : EventArgs);

#### *Description*

Represents the method that will handle the property builder event. The source of the event. An **System.EventArgs** object that provides data about the event.



## OnStylesChanged

[C#] protected internal void OnStylesChanged();  
[C++] protected public: void OnStylesChanged();  
[VB] Protected Friend Dim Sub OnStylesChanged()  
[JScript] package function OnStylesChanged();

### *Description*

Provides a method that can be overridden to implement functionality that should occur when a style of the control has been changed.

This method can serve as notification that a style of the control has changed.

## OnTemplateEditingVerbsChanged

[C#] protected abstract void OnTemplateEditingVerbsChanged();  
[C++] protected: virtual void OnTemplateEditingVerbsChanged() = 0;  
[VB] MustOverride Protected Sub OnTemplateEditingVerbsChanged()  
[JScript] protected abstract function OnTemplateEditingVerbsChanged();

### *Description*

Provides a method that can be overridden to implement functionality that should occur when the designer's template editing verbs have changed.

This method can serve as notification that the designer's template editing verbs have changed.

## PreFilterProperties

[C#] protected override void PreFilterProperties(IDictionary properties);

[C++] protected: void PreFilterProperties(IDictionary\* properties);

[VB] Overrides Protected Sub PreFilterProperties(ByVal properties As  
IDictionary)

[JScript] protected override function PreFilterProperties(properties : IDictionary);

#### *Description*

Filters the properties exposed through a  
**System.ComponentModel.TypeDescriptor** and replaces the runtime DataSource  
property descriptor with a design-time data source. The set of properties to filter.

BaseValidatorDesigner class (System.Web.UI.Design.WebControls)

UpdateDesignTimeHtml

#### *Description*

Provides design-time support for controls that derive from  
**System.Web.UI.WebControls.BaseValidator**.

BaseValidatorDesigner

#### *Example Syntax:*

UpdateDesignTimeHtml

[C#] public BaseValidatorDesigner();

[C++] public: BaseValidatorDesigner();

```

1 [VB] Public Sub New()
2 [JScript] public function BaseValidatorDesigner();
3     AllowResize
4     AssociatedComponents
5     Behavior
6     Component
7     DataBindings
8     DesignTimeElement
9     DesignTimeElementView
10    DesignTimeHtmlRequiresLoadComplete
11    ID
12    InheritanceAttribute
13    Inherited
14    IsDirty
15    ReadOnly
16    ShadowProperties
17    ShouldCodeSerialize
18    Verbs
19    GetDesignTimeHtml
20
21 [C#] public override string GetDesignTimeHtml();
22 [C++] public: String* GetDesignTimeHtml();
23 [VB] Overrides Public Function GetDesignTimeHtml() As String
24 [JScript] public override function GetDesignTimeHtml() : String;
25

```

1  
2 *Description*

3 Gets the HTML that is used to represent the control at design time.

4 *Return Value:* The HTML that is used to represent the control at design time.

5 ButtonDesigner class (System.Web.UI.Design.WebControls)

6 UpdateDesignTimeHtml

7  
8  
9 *Description*

10 Provides design-time support for the

11 **System.Web.UI.WebControls.Button** Web server control.

12 ButtonDesigner

13 *Example Syntax:*

14 UpdateDesignTimeHtml

15  
16 [C#] public ButtonDesigner();

17 [C++] public: ButtonDesigner();

18 [VB] Public Sub New()

19 [JScript] public function ButtonDesigner();

20 AllowResize

21 AssociatedComponents

22 Behavior

23 Component

24 DataBindings

25 DesignTimeElement

1 DesignTimeElementView  
 2 DesignTimeHtmlRequiresLoadComplete  
 3 ID  
 4 InheritanceAttribute  
 5 Inherited  
 6 IsDirty  
 7 ReadOnly  
 8 ShadowProperties  
 9 ShouldCodeSerialize  
 10 Verbs  
 11 GetDesignTimeHtml

12  
 13 [C#] public override string GetDesignTimeHtml();  
 14 [C++] public: String\* GetDesignTimeHtml();  
 15 [VB] Overrides Public Function GetDesignTimeHtml() As String  
 16 [JScript] public override function GetDesignTimeHtml() : String;  
 17

18 *Description*

19 Gets the HTML that is used to represent the control at design time.

20 *Return Value:* The HTML that is used to represent the control at design time.

21 CalendarAutoFormatDialog class (System.Web.UI.Design.WebControls)

22 UpdateDesignTimeHtml

23  
 24  
 25 *Description*

Provides an AutoFormat page for a  
**System.Web.UI.WebControls.Calendar** Web server control.

CalendarAutoFormatDialog

*Example Syntax:*

UpdateDesignTimeHtml

[C#] public CalendarAutoFormatDialog(Calendar calendar);

[C++] public: CalendarAutoFormatDialog(Calendar\* calendar);

[VB] Public Sub New(ByVal calendar As Calendar)

[JScript] public function CalendarAutoFormatDialog(calendar : Calendar);

### *Description*

Create a new AutoFormatPage instance Create a new AutoFormatPage  
instance

AcceptButton

AccessibilityObject

AccessibleDefaultActionDescription

AccessibleDescription

AccessibleName

AccessibleRole

ActiveControl

ActiveMdiChild

AllowDrop

AllowTransparency

Anchor

1	AutoScale
2	AutoScaleBaseSize
3	AutoScroll
4	AutoScrollMargin
5	AutoScrollMinSize
6	AutoScrollPosition
7	BackColor
8	BackgroundImage
9	BindingContext
10	Bottom
11	Bounds
12	CancelButton
13	CanFocus
14	CanSelect
15	Capture
16	CausesValidation
17	ClientRectangle
18	ClientSize
19	CompanyName
20	Container
21	ContainsFocus
22	ContextMenu
23	ControlBox
24	Controls
25	Created

1007420-00000000

1	CreateParams
2	Cursor
3	DataBindings
4	DefaultImeMode
5	DefaultSize
6	DesignMode
7	DesktopBounds
8	DesktopLocation
9	DialogResult
10	DisplayRectangle
11	Disposing
12	Dock
13	DockPadding
14	Enabled
15	Events
16	Focused
17	Font
18	FontHeight
19	ForeColor
20	FormBorderStyle
21	Handle
22	HasChildren
23	Height
24	HelpButton
25	HScroll



100720-6082660

1	Icon
2	ImeMode
3	InvokeRequired
4	IsAccessible
5	IsDisposed
6	IsHandleCreated
7	IsMdiChild
8	IsMdiContainer
9	IsRestrictedWindow
10	KeyPreview
11	Left
12	Location
13	MaximizeBox
14	MaximizedBounds
15	MaximumSize
16	MdiChildren
17	MdiParent
18	Menu
19	MergedMenu
20	MinimizeBox
21	MinimumSize
22	Modal
23	Name
24	Opacity
25	OwnedForms

1	Owner
2	Parent
3	ParentForm
4	ProductName
5	ProductVersion
6	RecreatingHandle
7	Region
8	RenderRightToLeft
9	ResizeRedraw
10	Right
11	RightToLeft
12	ShowFocusCues
13	ShowInTaskbar
14	ShowKeyboardCues
15	Site
16	Size
17	SizeGripStyle
18	StartPosition
19	TabIndex
20	TabStop
21	Tag
22	Text
23	Top
24	TopLevel
25	TopLevelControl

1 TopMost  
 2 TransparencyKey  
 3 Visible  
 4 VScroll  
 5 Width  
 6 WindowState  
 7 WindowTarget  
 8 DoDelayLoadActions

9  
 10 [C#] protected void DoDelayLoadActions();

11 [C++] protected: void DoDelayLoadActions();

12 [VB] Protected Sub DoDelayLoadActions()

13 [JScript] protected function DoDelayLoadActions();

14  
 15 *Description*

16 Executes any initialization that was delayed until the first idle time

17 Executes any initialization that was delayed until the first idle time

18 OnActivated

19  
 20 [C#] protected void OnActivated(object source, EventArgs e);

21 [C++] protected: void OnActivated(Object\* source, EventArgs\* e);

22 [VB] Protected Sub OnActivated(ByVal source As Object, ByVal e As  
 23 EventArgs)

24 [JScript] protected function OnActivated(source : Object, e : EventArgs);

25

1  
2 *Description*

3 Handles the activated event of the  
4 **System.Web.UI.WebControls.Calendar** AutoFormat dialog.

5 This method provides an opportunity to perform operations after a Calendar  
6 AutoFormat dialog has been activated. The source of the event. An  
7 **System.EventArgs** that provides information about the event.

8 **OnOKClicked**

9  
10 [C#] protected void OnOKClicked(object source, EventArgs e);  
11 [C++] protected: void OnOKClicked(Object\* source, EventArgs\* e);  
12 [VB] Protected Sub OnOKClicked(ByVal source As Object, ByVal e As  
13 EventArgs)  
14 [JScript] protected function OnOKClicked(source : Object, e : EventArgs);  
15

16 *Description*

17 Handle changes in the pre-defined schema choices Handle changes in the  
18 pre-defined schema choices

19 **OnSelChangedScheme**

20  
21 [C#] protected void OnSelChangedScheme(object source, EventArgs e);  
22 [C++] protected: void OnSelChangedScheme(Object\* source, EventArgs\* e);  
23 [VB] Protected Sub OnSelChangedScheme(ByVal source As Object, ByVal e As  
24 EventArgs)  
25 [JScript] protected function OnSelChangedScheme(source : Object, e :

EventArgs);

*Description*

Handle changes in the pre-defined schema choices Handle changes in the pre-defined schema choices

SaveComponent

[C#] protected void SaveComponent();

[C++] protected: void SaveComponent();

[VB] Protected Sub SaveComponent()

[JScript] protected function SaveComponent();

*Description*

Save any changes into the component Save any changes into the component

CalendarDesigner class (System.Web.UI.Design.WebControls)

WndProc

*Description*

Provides design-time support for the

**System.Web.UI.WebControls.Calendar** Web server control.

CalendarDesigner

*Example Syntax:*

WndProc

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25

- [C#] public CalendarDesigner();
- [C++] public: CalendarDesigner();
- [VB] Public Sub New()
- [JScript] public function CalendarDesigner();
- AllowResize
- AssociatedComponents
- Behavior
- Component
- DataBindings
- DesignTimeElement
- DesignTimeElementView
- DesignTimeHtmlRequiresLoadComplete
- ID
- InheritanceAttribute
- Inherited
- IsDirty
- ReadOnly
- ShadowProperties
- ShouldCodeSerialize
- Verbs
- WndProc

### Description

Gets the set of verbs available to this designer.

## Initialize

[C#] public override void Initialize(IComponent component);

[C++] public: void Initialize(IComponent\* component);

[VB] Overrides Public Sub Initialize(ByVal component As IComponent)

[JScript] public override function Initialize(component : IComponent);

### *Description*

Initializes the designer with the component for design.

The designer host calls

**System.Web.UI.Design.WebControls.CalendarDesigner.Initialize(System.ComponentModel.IComponent)** to establish the component for design. The control element for design.

## OnAutoFormat

[C#] protected void OnAutoFormat(object sender, EventArgs e);

[C++] protected: void OnAutoFormat(Object\* sender, EventArgs\* e);

[VB] Protected Sub OnAutoFormat(ByVal sender As Object, ByVal e As EventArgs)

[JScript] protected function OnAutoFormat(sender : Object, e : EventArgs);

### *Description*

Delegate to handle the the AutoFormat verb that displays a new

**System.Web.UI.Design.WebControls.CalendarAutoFormatDialog** . The

1 source of the event. The **System.EventArgs** object that provides data about the  
2 event.

3 **CheckBoxDesigner** class (System.Web.UI.Design.WebControls)

4 **UpdateDesignTimeHtml**

5  
6  
7 *Description*

8 Provides design-time support for the  
9 **System.Web.UI.WebControls.CheckBox** Web server control.

10 **CheckBoxDesigner**

11 *Example Syntax:*

12 **UpdateDesignTimeHtml**

13  
14 [C#] public **CheckBoxDesigner**();

15 [C++] public: **CheckBoxDesigner**();

16 [VB] Public Sub New()

17 [JScript] public function **CheckBoxDesigner**();

18 **AllowResize**

19 **AssociatedComponents**

20 **Behavior**

21 **Component**

22 **DataBindings**

23 **DesignTimeElement**

24 **DesignTimeElementView**

25 **DesignTimeHtmlRequiresLoadComplete**



- 1 ID
- 2 InheritanceAttribute
- 3 Inherited
- 4 IsDirty
- 5 ReadOnly
- 6 ShadowProperties
- 7 ShouldCodeSerialize
- 8 Verbs
- 9 GetDesignTimeHtml
- 10
- 11 [C#] public override string GetDesignTimeHtml();
- 12 [C++] public: String\* GetDesignTimeHtml();
- 13 [VB] Overrides Public Function GetDesignTimeHtml() As String
- 14 [JScript] public override function GetDesignTimeHtml() : String;
- 15

16 *Description*

17 Gets the HTML that is used to represent the control at design time.

18 *Return Value:* The HTML that is used to represent the control at design time.

19 DataGridColumnCollectionEditor class

20 (System.Web.UI.Design.WebControls)

21 UpdateDesignTimeHtml

22

23

24 *Description*

25

Provides a collection editing user interface for a column collection of a **System.Web.UI.WebControls.DataGrid** Web server control.

**DataGridColumnCollectionEditor**

*Example Syntax:*

**UpdateDesignTimeHtml**

[C#] public DataGridColumnCollectionEditor();

[C++] public: DataGridColumnCollectionEditor();

[VB] Public Sub New()

[JScript] public function DataGridColumnCollectionEditor();

**EditValue**

[C#] public override object EditValue(ITypeDescriptorContext context, IServiceProvider provider, object value);

[C++] public: Object\* EditValue(ITypeDescriptorContext\* context, IServiceProvider\* provider, Object\* value);

[VB] Overrides Public Function EditValue(ByVal context As ITypeDescriptorContext, ByVal provider As IServiceProvider, ByVal value As Object) As Object

[JScript] public override function EditValue(context : ITypeDescriptorContext, provider : IServiceProvider, value : Object) : Object;

### *Description*

Edits the specified value using the specified context descriptor and service provider.

1 *Return Value:* The new value. An

2 **System.ComponentModel.ITypeDescriptorContext** that indicates the context of  
3 the object to edit the value of. An **System.IServiceProvider**. The object to edit.

4       GetEditStyle

5  
6 [C#] public override UITypeEditorEditStyle

7 GetEditStyle(ITypeDescriptorContext context);

8 [C++] public: UITypeEditorEditStyle GetEditStyle(ITypeDescriptorContext\*  
9 context);

10 [VB] Overrides Public Function GetEditStyle(ByVal context As

11 ITypeDescriptorContext) As UITypeEditorEditStyle

12 [JScript] public override function GetEditStyle(context : ITypeDescriptorContext)  
13 : UITypeEditorEditStyle;

14  
15 *Description*

16       Gets the edit style used by the

17 **System.Web.UI.Design.WebControls.DataGridColumnCollectionEditor.EditValue**  
18 **Value(System.ComponentModel.ITypeDescriptorContext,System.IServicePro**  
19 **vider,System.Object)** method.

20 *Return Value:* A **System.Drawing.Design.UITypeEditorEditStyle** that  
21 represents the editor style that this editor uses. An

22 **System.ComponentModel.ITypeDescriptorContext** that indicates the context of  
23 the object to edit the value of.

24       DataGridColumnEditor class (System.Web.UI.Design.WebControls)

25       ToString

1  
2  
3 *Description*

4 Provides a component editor for a Web Forms

5 **System.Web.UI.WebControls.DataGrid** Web server control.

6 DataGridComponentEditor

7 *Example Syntax:*

8 ToString

9  
10 [C#] public DataGridComponentEditor();

11 [C++] public: DataGridComponentEditor();

12 [VB] Public Sub New()

13 [JScript] public function DataGridComponentEditor(); Initializes a new instance  
14 of the **System.Web.UI.Design.WebControls.DataGridComponentEditor** class.

15  
16 *Description*

17 Initializes a new instance of the

18 **System.Web.UI.Design.WebControls.DataGridComponentEditor** class.

19 DataGridComponentEditor

20 *Example Syntax:*

21 ToString

22  
23 [C#] public DataGridComponentEditor(int initialPage);

24 [C++] public: DataGridComponentEditor(int initialPage);

25 [VB] Public Sub New(ByVal initialPage As Integer)

1 [JScript] public function DataGridComponentEditor(initialPage : int);

2  
3 *Description*

4       Initializes a new instance of the  
5 **System.Web.UI.Design.WebControls.DataGridComponentEditor** class. The  
6 index of the initial page.

7       GetComponentEditorPages

8  
9 [C#] protected override Type[] GetComponentEditorPages();

10 [C++] protected: Type\* GetComponentEditorPages() [];

11 [VB] Overrides Protected Function GetComponentEditorPages() As Type()

12 [JScript] protected override function GetComponentEditorPages() : Type[];

13  
14 *Description*

15       Gets the set of all pages in the **System.Web.UI.WebControls.DataGrid**  
16 control.

17 *Return Value:* An array consisting of the set of component editor pages.

18       This method can be overridden to change the set of pages to show.

19       DataGridDesigner class (System.Web.UI.Design.WebControls)

20       ToString

21  
22  
23 *Description*

24       Provides design-time support for the  
25 **System.Web.UI.WebControls.DataGrid** Web server control.

1      DataGridDesigner

2      *Example Syntax:*

3      ToString

5      [C#] public DataGridDesigner();

6      [C++] public: DataGridDesigner();

7      [VB] Public Sub New()

8      [JScript] public function DataGridDesigner();

10     *Description*

11         Initializes a new instance of the

12     **System.Web.UI.Design.WebControls.DataGridDesigner** class.

13         ActiveTemplateEditingFrame

14         AllowResize

15         AssociatedComponents

16         Behavior

17         CanEnterTemplateMode

18         Component

19         DataBindings

20         DataKeyField

21         DataMember

22         DataSource

23         DesignTimeElement

24         DesignTimeElementView

25         DesignTimeHtmlRequiresLoadComplete

100720 6062060

1 HidePropertiesInTemplateMode  
2 ID  
3 InheritanceAttribute  
4 Inherited  
5 InTemplateMode  
6 IsDirty  
7 ReadOnly  
8 ShadowProperties  
9 ShouldCodeSerialize  
10 Verbs  
11 CreateTemplateEditingFrame

12  
13 [C#] protected override ITemplateEditingFrame  
14 CreateTemplateEditingFrame(TemplateEditingVerb verb);  
15 [C++] protected: ITemplateEditingFrame\*  
16 CreateTemplateEditingFrame(TemplateEditingVerb\* verb);  
17 [VB] Overrides Protected Function CreateTemplateEditingFrame(ByVal verb As  
18 TemplateEditingVerb) As ITemplateEditingFrame  
19 [JScript] protected override function CreateTemplateEditingFrame(verb :  
20 TemplateEditingVerb) : ITemplateEditingFrame;

21  
22 *Description*

23 Creates a template editing frame using the specified verb.

24 *Return Value:* The new template editing frame. The verb to create the template  
25 editing frame for.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

Dispose

```
[C#] protected override void Dispose(bool disposing);  
[C++] protected: void Dispose(bool disposing);  
[VB] Overrides Protected Sub Dispose(ByVal disposing As Boolean)  
[JScript] protected override function Dispose(disposing : Boolean);
```

*Description*

Releases the unmanaged resources used by the **System.Web.UI.Design.WebControls.DataGridDesigner** and optionally releases the managed resources.

This method is called by the public **Dispose()** method and the **System.Object.Finalize** method. **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

GetCachedTemplateEditingVerbs

```
[C#] protected override TemplateEditingVerb[]  
GetCachedTemplateEditingVerbs();  
[C++] protected: TemplateEditingVerb* GetCachedTemplateEditingVerbs() [];  
[VB] Overrides Protected Function GetCachedTemplateEditingVerbs() As  
TemplateEditingVerb()  
[JScript] protected override function GetCachedTemplateEditingVerbs() :  
TemplateEditingVerb[];
```

*Description*



1 Gets the cached template editing verbs.  
 2 *Return Value:* An array containing the cached template editing verbs.

3 GetDesignTimeHtml

4  
 5 [C#] public override string GetDesignTimeHtml();  
 6 [C++] public: String\* GetDesignTimeHtml();  
 7 [VB] Overrides Public Function GetDesignTimeHtml() As String  
 8 [JScript] public override function GetDesignTimeHtml() : String;

9  
 10 *Description*

11 Gets the HTML that is used to represent the control at design time.  
 12 *Return Value:* The HTML that is used to represent the control at design time.

13 GetEmptyDesignTimeHtml

14  
 15 [C#] protected override string GetEmptyDesignTimeHtml();  
 16 [C++] protected: String\* GetEmptyDesignTimeHtml();  
 17 [VB] Overrides Protected Function GetEmptyDesignTimeHtml() As String  
 18 [JScript] protected override function GetEmptyDesignTimeHtml() : String;

19  
 20 *Description*

21 Gets the HTML used to represent an empty template-based control at  
 22 design time.  
 23 *Return Value:* The HTML used to represent an empty template-based control at  
 24 design time.

25 GetErrorDesignTimeHtml

1 [C#] protected override string GetErrorDesignTimeHtml(Exception e);

2 [C++] protected: String\* GetErrorDesignTimeHtml(Exception\* e);

3 [VB] Overrides Protected Function GetErrorDesignTimeHtml(ByVal e As  
4 Exception) As String

5 [JScript] protected override function GetErrorDesignTimeHtml(e : Exception) :  
6 String;

7  
8  
9 *Description*

10 Gets the HTML displayed at design-time for the specified exception when  
11 an error has been encountered while rendering the control.

12 *Return Value:* The HTML displayed at design-time for the specified exception.

13 The exception to display an error message for.

14 *GetTemplateContainerDataItemProperty*

15  
16 [C#] public override string GetTemplateContainerDataItemProperty(string  
17 templateName);

18 [C++] public: String\* GetTemplateContainerDataItemProperty(String\*  
19 templateName);

20 [VB] Overrides Public Function GetTemplateContainerDataItemProperty(ByVal  
21 templateName As String) As String

22 [JScript] public override function

23 GetTemplateContainerDataItemProperty(templateName : String) : String;

24  
25 *Description*

Gets the template's container's data item property.

*Return Value:* The template's container's data item property. The name of the template.

#### GetTemplateContent

[C#] public override string GetTemplateContent(ITemplateEditingFrame editingFrame, string templateName, out bool allowEditing);

[C++] public: String\* GetTemplateContent(ITemplateEditingFrame\* editingFrame, String\* templateName, bool\* allowEditing);

[VB] Overrides Public Function GetTemplateContent(ByVal editingFrame As ITemplateEditingFrame, ByVal templateName As String, ByRef allowEditing As Boolean) As String

[JScript] public override function GetTemplateContent(editingFrame : ITemplateEditingFrame, templateName : String, allowEditing : Boolean) : String;

#### *Description*

Gets the template's content.

*Return Value:* The template's content. The template frame to retrieve content for. The name of the template. A boolean variable that will be set to **true** if the template's content can be edited, or **false** if the content is read-only.

#### GetTemplatePropertyParentType

[C#] public override Type GetTemplatePropertyParentType(string templateName);

[C++] public: Type\* GetTemplatePropertyParentType(String\* templateName);

1 [VB] Overrides Public Function GetTemplatePropertyParentType(ByVal  
2 templateName As String) As Type  
3 [JScript] public override function GetTemplatePropertyParentType(templateName  
4 : String) : Type;

5  
6 *Description*

7 Gets the type of the parent of the template property.

8 *Return Value:* The type of the object that has the template property. The name of  
9 the template to return the type of the parent for.

10 Initialize

11  
12 [C#] public override void Initialize(IComponent component);

13 [C++] public: void Initialize(IComponent\* component);

14 [VB] Overrides Public Sub Initialize(ByVal component As IComponent)

15 [JScript] public override function Initialize(component : IComponent);

16  
17 *Description*

18 Initializes the designer with the **System.Web.UI.WebControls.DataGrid**  
19 control that this instance of the designer is associated with. The associated  
20 **System.Web.UI.WebControls.DataGrid** control.

21 OnColumnsChanged

22  
23 [C#] public virtual void OnColumnsChanged();

24 [C++] public: virtual void OnColumnsChanged();

25 [VB] Overridable Public Sub OnColumnsChanged()

1 [JScript] public function OnColumnsChanged();

3 *Description*

4 Notification that is called when the columns changed event occurs.

5 OnTemplateEditingVerbsChanged

7 [C#] protected override void OnTemplateEditingVerbsChanged();

8 [C++] protected: void OnTemplateEditingVerbsChanged();

9 [VB] Overrides Protected Sub OnTemplateEditingVerbsChanged()

10 [JScript] protected override function OnTemplateEditingVerbsChanged();

12 *Description*

13 Provides an opportunity to do processing or other actions when a change  
14 has been made to the template editing verbs.

15 SetTemplateContent

17 [C#] public override void SetTemplateContent(ITemplateEditingFrame  
18 editingFrame, string templateName, string templateContent);

19 [C++] public: void SetTemplateContent(ITemplateEditingFrame\* editingFrame,  
20 String\* templateName, String\* templateContent);

21 [VB] Overrides Public Sub SetTemplateContent(ByVal editingFrame As  
22 ITemplateEditingFrame, ByVal templateName As String, ByVal templateContent  
23 As String)

24 [JScript] public override function SetTemplateContent(editingFrame :  
25 ITemplateEditingFrame, templateName : String, templateContent : String);

1  
2 *Description*

3       Sets the content for the specified template and frame. The template frame to  
4 set the content for. The name of the template. The content to set.

5       DataListComponentEditor class (System.Web.UI.Design.WebControls)  
6       UpdateDesignTimeHtml  
7  
8

9 *Description*

10       Provides a component editor for a Web Forms  
11 **System.Web.UI.WebControls.DataList** control.

12       DataListComponentEditor

13 *Example Syntax:*

14       UpdateDesignTimeHtml  
15

16 [C#] public DataListComponentEditor();

17 [C++] public: DataListComponentEditor();

18 [VB] Public Sub New()

19 [JScript] public function DataListComponentEditor(); Initializes a new instance of  
20 **System.Web.UI.Design.WebControls.DataListComponentEditor** .  
21

22 *Description*

23       Initializes a new instance of  
24 **System.Web.UI.Design.WebControls.DataListComponentEditor** .

25       DataListComponentEditor

*Example Syntax:*

UpdateDesignTimeHtml

```
[C#] public DataListComponentEditor(int initialPage);  
[C++] public: DataListComponentEditor(int initialPage);  
[VB] Public Sub New(ByVal initialPage As Integer)  
[JScript] public function DataListComponentEditor(initialPage : int);
```

*Description*

Initializes a new instance of **System.Web.UI.Design.WebControls.DataListComponentEditor**. The index of the initial page to display.

GetComponentEditorPages

```
[C#] protected override Type[] GetComponentEditorPages();  
[C++] protected: Type* GetComponentEditorPages() [];  
[VB] Overrides Protected Function GetComponentEditorPages() As Type()  
[JScript] protected override function GetComponentEditorPages() : Type[];
```

*Description*

Gets the set of component editor pages owned by the designer.

*Return Value:* The pages owned by the designer.

DataListDesigner class (System.Web.UI.Design.WebControls)

ToString

1  
2  
3 *Description*

4 Provides design-time support for the  
5 **System.Web.UI.WebControls.DataList** Web server control.

6 DataListDesigner

7 *Example Syntax:*

8 ToString

9  
10 [C#] public DataListDesigner();

11 [C++] public: DataListDesigner();

12 [VB] Public Sub New()

13 [JScript] public function DataListDesigner();

14  
15 *Description*

16 Initializes a new instance of the  
17 **System.Web.UI.Design.WebControls.DataListDesigner** class.

18 ActiveTemplateEditingFrame

19 AllowResize

20 ToString

21  
22  
23 *Description*

24 Indicates whether the data list can be resized.

25 AssociatedComponents



1	Behavior
2	CanEnterTemplateMode
3	Component
4	DataBindings
5	DataKeyField
6	DataMember
7	DataSource
8	DesignTimeElement
9	DesignTimeElementView
10	DesignTimeHtmlRequiresLoadComplete
11	HidePropertiesInTemplateMode
12	ID
13	InheritanceAttribute
14	Inherited
15	InTemplateMode
16	IsDirty
17	ReadOnly
18	ShadowProperties
19	ShouldCodeSerialize
20	TemplatesExist
21	ToString

24 *Description*

25 Indicates whether templates associated to the designer currently exist.

## Verbs

### CreateTemplateEditingFrame

[C#] protected override ITemplateEditingFrame

CreateTemplateEditingFrame(TemplateEditingVerb verb);

[C++] protected: ITemplateEditingFrame\*

CreateTemplateEditingFrame(TemplateEditingVerb\* verb);

[VB] Overrides Protected Function CreateTemplateEditingFrame(ByVal verb As

TemplateEditingVerb) As ITemplateEditingFrame

[JScript] protected override function CreateTemplateEditingFrame(verb :

TemplateEditingVerb) : ITemplateEditingFrame;

#### *Description*

Creates a template editing frame using the specified verb.

**Return Value:** An **System.Web.UI.Design.ITemplateEditingFrame** . The verb that was invoked to create a template editing frame.

### Dispose

[C#] protected override void Dispose(bool disposing);

[C++] protected: void Dispose(bool disposing);

[VB] Overrides Protected Sub Dispose(ByVal disposing As Boolean)

[JScript] protected override function Dispose(disposing : Boolean);

#### *Description*

Releases the unmanaged resources used by the **System.Web.UI.Design.WebControls.DataListDesigner** and optionally releases the managed resources.

This method is called by the public **Dispose()** method and the **System.Object.Finalize** method. **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

GetCachedTemplateEditingVerbs

[C#] protected override TemplateEditingVerb[]

GetCachedTemplateEditingVerbs();

[C++] protected: TemplateEditingVerb\* GetCachedTemplateEditingVerbs() [];

[VB] Overrides Protected Function GetCachedTemplateEditingVerbs() As

TemplateEditingVerb()

[JScript] protected override function GetCachedTemplateEditingVerbs() :

TemplateEditingVerb[];

### *Description*

Gets the cached template editing verbs available to this designer.

*Return Value:* The cached template editing verbs available to this designer.

GetDesignTimeHtml

[C#] public override string GetDesignTimeHtml();

[C++] public: String\* GetDesignTimeHtml();

[VB] Overrides Public Function GetDesignTimeHtml() As String

[JScript] public override function GetDesignTimeHtml() : String;

1  
2 *Description*

3 Gets the HTML that is used to represent the control at design time.

4 *Return Value:* The HTML that is used to represent the control at design time.

5 GetEmptyDesignTimeHtml

6  
7 [C#] protected override string GetEmptyDesignTimeHtml();

8 [C++] protected: String\* GetEmptyDesignTimeHtml();

9 [VB] Overrides Protected Function GetEmptyDesignTimeHtml() As String

10 [JScript] protected override function GetEmptyDesignTimeHtml() : String;

11  
12 *Description*

13 Gets the HTML used to represent an empty template-based control at  
14 design time.

15 *Return Value:* The HTML used to represent an empty template-based control at  
16 design time.

17 GetErrorDesignTimeHtml

18  
19 [C#] protected override string GetErrorDesignTimeHtml(Exception e);

20 [C++] protected: String\* GetErrorDesignTimeHtml(Exception\* e);

21 [VB] Overrides Protected Function GetErrorDesignTimeHtml(ByVal e As  
22 Exception) As String

23 [JScript] protected override function GetErrorDesignTimeHtml(e : Exception) :  
24 String;

1  
2 *Description*

3 Gets the HTML to display at design-time for the specified exception when  
4 an error has been encountered while rendering the control.

5 *Return Value:* The HTML displayed at design-time for the specified exception.  
6 The exception to display the error message for.

7 **GetTemplateContainerDataItemProperty**

8  
9 [C#] public override string GetTemplateContainerDataItemProperty(string  
10 templateName);

11 [C++] public: String\* GetTemplateContainerDataItemProperty(String\*  
12 templateName);

13 [VB] Overrides Public Function GetTemplateContainerDataItemProperty(ByVal  
14 templateName As String) As String

15 [JScript] public override function

16 GetTemplateContainerDataItemProperty(templateName : String) : String;

17  
18 *Description*

19 Gets the template's container's data item property.

20 *Return Value:* The data item property of the container of the template. The name  
21 of the template.

22 **GetTemplateContent**

23  
24 [C#] public override string GetTemplateContent(ITemplateEditingFrame  
25 editingFrame, string templateName, out bool allowEditing);

```

1 [C++] public: String* GetTemplateContent(ITemplateEditingFrame*
2 editingFrame, String* templateName, bool* allowEditing);
3 [VB] Overrides Public Function GetTemplateContent(ByVal editingFrame As
4 ITemplateEditingFrame, ByVal templateName As String, ByRef allowEditing As
5 Boolean) As String
6 [JScript] public override function GetTemplateContent(editingFrame :
7 ITemplateEditingFrame, templateName : String, allowEditing : Boolean) : String;
8

```

### *Description*

Gets the template's content.

**Return Value:** The template's content. The template frame to get content from. The name of the template. A boolean variable that will be set to **true** if the template's content can be edited, or **false** if the content is read-only.

### *Initialize*

```

14 Initialize
15
16 [C#] public override void Initialize(IComponent component);
17 [C++] public: void Initialize(IComponent* component);
18 [VB] Overrides Public Sub Initialize(ByVal component As IComponent)
19 [JScript] public override function Initialize(component : IComponent);
20

```

### *Description*

Initializes the designer with the **System.Web.UI.WebControls.DataList** control that this instance of the designer is associated with. The associated **System.Web.UI.WebControls.DataList** control.

### *OnTemplateEditingVerbsChanged*

1  
2 [C#] protected override void OnTemplateEditingVerbsChanged();  
3 [C++] protected: void OnTemplateEditingVerbsChanged();  
4 [VB] Overrides Protected Sub OnTemplateEditingVerbsChanged()  
5 [JScript] protected override function OnTemplateEditingVerbsChanged();  
6

### 7 *Description*

8 Provides an opportunity to do processing or other actions when a change  
9 has been made to the template editing verbs.

### 10 *SetTemplateContent*

11  
12 [C#] public override void SetTemplateContent(ITemplateEditingFrame  
13 editingFrame, string templateName, string templateContent);  
14 [C++] public: void SetTemplateContent(ITemplateEditingFrame\* editingFrame,  
15 String\* templateName, String\* templateContent);  
16 [VB] Overrides Public Sub SetTemplateContent(ByVal editingFrame As  
17 ITemplateEditingFrame, ByVal templateName As String, ByVal templateContent  
18 As String)  
19 [JScript] public override function SetTemplateContent(editingFrame :  
20 ITemplateEditingFrame, templateName : String, templateContent : String);  
21

### 22 *Description*

23 Sets the template's content. The template frame to set content for. The name  
24 of the template. The content to set.

25 HyperLinkDesigner class (System.Web.UI.Design.WebControls)

## UpdateDesignTimeHtml

### *Description*

Provides design-time support for the  
**System.Web.UI.WebControls.HyperLink** Web server control.

HyperLinkDesigner

### *Example Syntax:*

UpdateDesignTimeHtml

[C#] public HyperLinkDesigner();

[C++] public: HyperLinkDesigner();

[VB] Public Sub New()

[JScript] public function HyperLinkDesigner();

AllowResize

AssociatedComponents

Behavior

Component

DataBindings

DesignTimeElement

DesignTimeElementView

DesignTimeHtmlRequiresLoadComplete

ID

InheritanceAttribute

Inherited



1 IsDirty  
 2 ReadOnly  
 3 ShadowProperties  
 4 ShouldCodeSerialize  
 5 Verbs  
 6 GetDesignTimeHtml  
 7  
 8 [C#] public override string GetDesignTimeHtml();  
 9 [C++] public: String\* GetDesignTimeHtml();  
 10 [VB] Overrides Public Function GetDesignTimeHtml() As String  
 11 [JScript] public override function GetDesignTimeHtml() : String;  
 12

### 13 *Description*

14 Gets the HTML that is used to represent the control at design time.

15 *Return Value:* The HTML that is used to represent the control at design time.

16 LabelDesigner class (System.Web.UI.Design.WebControls)

17 UpdateDesignTimeHtml  
 18  
 19

### 20 *Description*

21 Provides design-time support for the **System.Web.UI.WebControls.Label**

22 Web server control.

23 LabelDesigner

24 *Example Syntax:*

25 UpdateDesignTimeHtml

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#] public LabelDesigner();
[C++] public: LabelDesigner();
[VB] Public Sub New()
[JavaScript] public function LabelDesigner();

    AllowResize
    AssociatedComponents
    Behavior
    Component
    DataBindings
    DesignTimeElement
    DesignTimeElementView
    DesignTimeHtmlRequiresLoadComplete
    ID
    InheritanceAttribute
    Inherited
    IsDirty
    ReadOnly
    ShadowProperties
    ShouldCodeSerialize
    Verbs

    LinkButtonDesigner class (System.Web.UI.Design.WebControls)
    UpdateDesignTimeHtml
```

1  
2  
3 *Description*

4 Provides design-time support for the  
5 **System.Web.UI.WebControls.LinkButton** Web server control.

6 LinkButtonDesigner

7 *Example Syntax:*

8 UpdateDesignTimeHtml

9  
10 [C#] public LinkButtonDesigner();

11 [C++] public: LinkButtonDesigner();

12 [VB] Public Sub New()

13 [JScript] public function LinkButtonDesigner();

14 AllowResize

15 AssociatedComponents

16 Behavior

17 Component

18 DataBindings

19 DesignTimeElement

20 DesignTimeElementView

21 DesignTimeHtmlRequiresLoadComplete

22 ID

23 InheritanceAttribute

24 Inherited

25 IsDirty

1      ReadOnly  
 2      ShadowProperties  
 3      ShouldCodeSerialize  
 4      Verbs  
 5      ListControlDataBindingHandler class  
 6      (System.Web.UI.Design.WebControls)  
 7      UpdateDesignTimeHtml

8  
 9  
 10    *Description*

11      Provides a data binding handler for a  
 12      **System.Web.UI.WebControls.ListControl** .

13      ListControlDataBindingHandler

14      *Example Syntax:*

15      UpdateDesignTimeHtml

16  
 17    [C#] public ListControlDataBindingHandler();

18    [C++] public: ListControlDataBindingHandler();

19    [VB] Public Sub New()

20    [JScript] public function ListControlDataBindingHandler();

21      DataBindControl

22  
 23    [C#] public override void DataBindControl(IDesignerHost designerHost, Control  
 24    control);

25    [C++] public: void DataBindControl(IDesignerHost\* designerHost, Control\*

1 control);

2 [VB] Overrides Public Sub DataBindControl(ByVal designerHost As

3 IDesignerHost, ByVal control As Control)

4 [JScript] public override function DataBindControl(designerHost : IDesignerHost,

5 control : Control);

6  
7 *Description*

8 Adds a data binding to the specified control. The designer host for the  
9 document that contains the control. The control to add this data binding to.

10 ListControlDesigner class (System.Web.UI.Design.WebControls)

11 ToString

12  
13  
14 *Description*

15 Provides design-time support for  
16 **System.Web.UI.WebControls.ListControl** Web server controls.

17 ListControlDesigner

18 *Example Syntax:*

19 ToString

20  
21 [C#] public ListControlDesigner();

22 [C++] public: ListControlDesigner();

23 [VB] Public Sub New()

24 [JScript] public function ListControlDesigner();

1  
2 *Description*

3       Initializes a new instance of the  
4 **System.Web.UI.Design.WebControls.ListControlDesigner** class.

5       AllowResize

6       AssociatedComponents

7       Behavior

8       Component

9       DataBindings

10      DataMember

11      ToString

12  
13  
14 *Description*

15       Gets or sets the data member for the control.

16       DataSource

17       ToString

18  
19 [C#] public string DataSource {get; set;}

20 [C++] public: \_\_property String\* get\_DataSource();public: \_\_property void  
21 set\_DataSource(String\*);

22 [VB] Public Property DataSource As String

23 [JScript] public function get DataSource() : String;public function set  
24 DataSource(String);

1  
2 *Description*

3 Gets or sets the data source property of the control.

4 DataTextField

5 ToString

6  
7 [C#] public string DataTextField {get; set;}

8 [C++] public: \_\_property String\* get\_DataTextField();public: \_\_property void  
9 set\_DataTextField(String\*);

10 [VB] Public Property DataTextField As String

11 [JScript] public function get DataTextField() : String;public function set  
12 DataTextField(String);

13  
14 *Description*

15 Gets or sets the data text field of the control.

16 DataValueField

17 ToString

18  
19 [C#] public string DataValueField {get; set;}

20 [C++] public: \_\_property String\* get\_DataValueField();public: \_\_property void  
21 set\_DataValueField(String\*);

22 [VB] Public Property DataValueField As String

23 [JScript] public function get DataValueField() : String;public function set  
24 DataValueField(String);

1  
2 *Description*

3 Gets or sets the data value field of the control.

4 DesignTimeElement

5 DesignTimeElementView

6 DesignTimeHtmlRequiresLoadComplete

7 ID

8 InheritanceAttribute

9 Inherited

10 IsDirty

11 ReadOnly

12 ShadowProperties

13 ShouldCodeSerialize

14 Verbs

15 GetDesignTimeHtml

16  
17 [C#] public override string GetDesignTimeHtml();

18 [C++] public: String\* GetDesignTimeHtml();

19 [VB] Overrides Public Function GetDesignTimeHtml() As String

20 [JScript] public override function GetDesignTimeHtml() : String;

21  
22 *Description*

23 Gets the HTML that is used to represent the control at design time.

24 *Return Value:* The HTML that is used to represent the control at design time.

25 GetResolvedSelectedDataSource



1  
2 [C#] public IEnumerable GetResolvedSelectedDataSource();

3 [C++] public: \_\_sealed IEnumerable\* GetResolvedSelectedDataSource();

4 [VB] NotOverridable Public Function GetResolvedSelectedDataSource() As  
5 IEnumerable

6 [JScript] public function GetResolvedSelectedDataSource() : IEnumerable;

7  
8 *Description*

9 Gets the data member that is currently selected within the data source  
10 currently bound to the control.

11 *Return Value:* The currently selected data member, or **null** if the control was not  
12 bound to a data source, or the data source, site of the designer's component, or the  
13 container of the data source could not be accessed.

14 GetSelectedDataSource

15  
16 [C#] public object GetSelectedDataSource();

17 [C++] public: \_\_sealed Object\* GetSelectedDataSource();

18 [VB] NotOverridable Public Function GetSelectedDataSource() As Object

19 [JScript] public function GetSelectedDataSource() : Object;

20  
21 *Description*

22 Gets the selected data source component from the component's container.

23 *Return Value:* The selected data source, or **null** if a data source is not found or if a  
24 data source with the same name does not exist.

25 Initialize

```

1 [C#] public override void Initialize(IComponent component);
2
3 [C++] public: void Initialize(IComponent* component);
4
5 [VB] Overrides Public Sub Initialize(ByVal component As IComponent)
6
7 [JScript] public override function Initialize(component : IComponent);
8

```

### *Description*

Initializes the component for design.

This method is called to initialize the designer with the component to design. The control that is being designed.

### *OnComponentChanged*

```

13 [C#] public override void OnComponentChanged(object source,
14 ComponentChangedEventArgs ce);
15
16 [C++] public: void OnComponentChanged(Object* source,
17 ComponentChangedEventArgs* ce);
18
19 [VB] Overrides Public Sub OnComponentChanged(ByVal source As Object,
20 ByVal ce As ComponentChangedEventArgs)
21
22 [JScript] public override function OnComponentChanged(source : Object, ce :
23 ComponentChangedEventArgs);
24

```

### *Description*

Raises the ComponentChanged event.

Handles changes made to the component. This includes changes made in the Properties window. The source of the event. A

1 **System.ComponentModel.Design.ComponentChangedEventArgs** that  
2 provides data about the event.

3       OnDataSourceChanged

4  
5 [C#] public virtual void OnDataSourceChanged();

6 [C++] public: virtual void OnDataSourceChanged();

7 [VB] Overridable Public Sub OnDataSourceChanged()

8 [JScript] public function OnDataSourceChanged();

9  
10 *Description*

11       Raises the DataSource event.

12       Handles changes made to the data source.

13       PreFilterProperties

14  
15 [C#] protected override void PreFilterProperties(IDictionary properties);

16 [C++] protected: void PreFilterProperties(IDictionary\* properties);

17 [VB] Overrides Protected Sub PreFilterProperties(ByVal properties As  
18 IDictionary)

19 [JScript] protected override function PreFilterProperties(properties : IDictionary);

20  
21 *Description*

22       Filters the properties exposed through a

23 **System.ComponentModel.TypeDescriptor** and replaces the property descriptor  
24 for the DataSource property, which contains the value used at runtime, with a

property descriptor that contains a value indicating a design-time data source. The properties of the control.

ListItemsCollectionEditor class (System.Web.UI.Design.WebControls)

UpdateDesignTimeHtml

### *Description*

Provides a user interface for editing the items collection of a list.

ListItemsCollectionEditor

### *Example Syntax:*

UpdateDesignTimeHtml

[C#] public ListItemsCollectionEditor(Type type);

[C++] public: ListItemsCollectionEditor(Type\* type);

[VB] Public Sub New(ByVal type As Type)

[JScript] public function ListItemsCollectionEditor(type : Type);

### *Description*

Initializes a new instance of the

**System.Web.UI.Design.WebControls.ListItemsCollectionEditor** class. The type of the collection to edit.

CollectionItemType

CollectionType

Context

HelpTopic

1       NewItemTypes  
 2       CanSelectMultipleInstances  
 3  
 4   [C#] protected override bool CanSelectMultipleInstances();  
 5   [C++] protected: bool CanSelectMultipleInstances();  
 6   [VB] Overrides Protected Function CanSelectMultipleInstances() As Boolean  
 7   [JScript] protected override function CanSelectMultipleInstances() : Boolean;

8  
 9   *Description*

10       Indicates whether multiple items in the list can be selected at the same time.

11   *Return Value:* **true** if multiple items can be selected at the same time; otherwise,  
 12   **false** .

13       PanelDesigner class (System.Web.UI.Design.WebControls)

14       ToString  
 15  
 16

17   *Description*

18       Provides design-time support for the **System.Web.UI.WebControls.Panel**

19   Web server control.

20       PanelDesigner

21       *Example Syntax:*

22       ToString  
 23

24   [C#] public PanelDesigner();

25   [C++] public: PanelDesigner();

```

1  [VB] Public Sub New()
2  [JScript] public function PanelDesigner();
3      AllowResize
4      AssociatedComponents
5      Behavior
6      Component
7      DataBindings
8      DesignTimeElement
9      DesignTimeElementView
10     DesignTimeHtmlRequiresLoadComplete
11     ID
12     InheritanceAttribute
13     Inherited
14     IsDirty
15     ReadOnly
16     ShadowProperties
17     ShouldCodeSerialize
18     Verbs
19     MapPropertyToStyle
20
21 [C#] protected override void MapPropertyToStyle(string propName, object
22 varPropValue);
23 [C++] protected: void MapPropertyToStyle(String* propName, Object*
24 varPropValue);
25 [VB] Overrides Protected Sub MapPropertyToStyle(ByVal propName As String,

```

1 ByVal varPropValue As Object)  
2 [JScript] protected override function MapPropertyToStyle(propName : String,  
3 varPropValue : Object);  
4

5 *Description*

6 Maps a specified property and value to a specified HTML style. The  
7 property name. The property value.

8 OnBehaviorAttached

9  
10 [C#] protected override void OnBehaviorAttached();

11 [C++] protected: void OnBehaviorAttached();

12 [VB] Overrides Protected Sub OnBehaviorAttached()

13 [JScript] protected override function OnBehaviorAttached();  
14

15 *Description*

16 Provides notification when a behavior is attached to the designer.

17 RegexEditorDialog class (System.Web.UI.Design.WebControls)

18 UpdateDesignTimeHtml  
19  
20

21 *Description*

22 Provides a dialog for editing regular expressions used by the

23 **System.Web.UI.WebControls.RegularExpressionValidator** .

24 RegexEditorDialog

25 *Example Syntax:*

UpdateDesignTimeHtml

[C#] public RegexEditorDialog(ISite site);

[C++] public: RegexEditorDialog(ISite\* site);

[VB] Public Sub New(ByVal site As ISite)

[JScript] public function RegexEditorDialog(site : ISite);

### *Description*

Initializes a new instance of the **System.Web.UI.Design.WebControls.RegexEditorDialog** class. The site for this dialog.

AcceptButton

AccessibilityObject

AccessibleDefaultActionDescription

AccessibleDescription

AccessibleName

AccessibleRole

ActiveControl

ActiveMdiChild

AllowDrop

AllowTransparency

Anchor

AutoScale

AutoScaleBaseSize

AutoScroll



1	AutoScrollMargin
2	AutoScrollMinSize
3	AutoScrollPosition
4	BackColor
5	BackgroundImage
6	BindingContext
7	Bottom
8	Bounds
9	CancelButton
10	CanFocus
11	CanSelect
12	Capture
13	CausesValidation
14	ClientRectangle
15	ClientSize
16	CompanyName
17	Container
18	ContainsFocus
19	ContextMenu
20	ControlBox
21	Controls
22	Created
23	CreateParams
24	Cursor
25	DataBindings

1	DefaultImeMode
2	DefaultSize
3	DesignMode
4	DesktopBounds
5	DesktopLocation
6	DialogResult
7	DisplayRectangle
8	Disposing
9	Dock
10	DockPadding
11	Enabled
12	Events
13	Focused
14	Font
15	FontHeight
16	ForeColor
17	FormBorderStyle
18	Handle
19	HasChildren
20	Height
21	HelpButton
22	HScroll
23	Icon
24	ImeMode
25	InvokeRequired

1	IsAccessible
2	IsDisposed
3	IsHandleCreated
4	IsMdiChild
5	IsMdiContainer
6	IsRestrictedWindow
7	KeyPreview
8	Left
9	Location
10	MaximizeBox
11	MaximizedBounds
12	MaximumSize
13	MdiChildren
14	MdiParent
15	Menu
16	MergedMenu
17	MinimizeBox
18	MinimumSize
19	Modal
20	Name
21	Opacity
22	OwnedForms
23	Owner
24	Parent
25	ParentForm

1 ProductName  
2 ProductVersion  
3 RecreatingHandle  
4 Region  
5 RegularExpression  
6 UpdateDesignTimeHtml

7  
8  
9 *Description*

10 Gets or sets the name of the regular expression to edit.  
11 RenderRightToLeft  
12 ResizeRedraw  
13 Right  
14 RightToLeft  
15 ShowFocusCues  
16 ShowInTaskbar  
17 ShowKeyboardCues  
18 Site  
19 Size  
20 SizeGripStyle  
21 StartPosition  
22 TabIndex  
23 TabStop  
24 Tag  
25 Text

1 Top  
2 TopLevel  
3 TopLevelControl  
4 TopMost  
5 TransparencyKey  
6 Visible  
7 VScroll  
8 Width  
9 WindowState  
10 WindowTarget  
11 cmdHelp\_Click

12  
13 [C#] protected void cmdHelp\_Click(object sender, EventArgs e);  
14 [C++] protected: void cmdHelp\_Click(Object\* sender, EventArgs\* e);  
15 [VB] Protected Sub cmdHelp\_Click(ByVal sender As Object, ByVal e As  
16 EventArgs)  
17 [JScript] protected function cmdHelp\_Click(sender : Object, e : EventArgs);  
18

19 *Description*

20 Represents the method that will handle the Help event of the dialog. The  
21 source of the event. An EventArgs that provides data for the event.

22 cmdOK\_Click

23  
24 [C#] protected void cmdOK\_Click(object sender, EventArgs e);  
25 [C++] protected: void cmdOK\_Click(Object\* sender, EventArgs\* e);

1 [VB] Protected Sub cmdOK\_Click(ByVal sender As Object, ByVal e As  
2 EventArgs)

3 [JScript] protected function cmdOK\_Click(sender : Object, e : EventArgs);

4  
5 *Description*

6 Represents the method that will handle the OK event of the dialog. The  
7 source of the event. An EventArgs that provides data for the event.

8 cmdTestValidate\_Click

9  
10 [C#] protected void cmdTestValidate\_Click(object sender, EventArgs args);

11 [C++] protected: void cmdTestValidate\_Click(Object\* sender, EventArgs\* args);

12 [VB] Protected Sub cmdTestValidate\_Click(ByVal sender As Object, ByVal args As  
13 EventArgs)

14 [JScript] protected function cmdTestValidate\_Click(sender : Object, args :  
15 EventArgs);

16  
17 *Description*

18 Represents the method that will handle the XXX event of a XXX. The  
19 source of the event. An EventArgs that provides data for the event.

20 Dispose

21  
22 [C#] protected override void Dispose(bool disposing);

23 [C++] protected: void Dispose(bool disposing);

24 [VB] Overrides Protected Sub Dispose(ByVal disposing As Boolean)

25 [JScript] protected override function Dispose(disposing : Boolean);

1  
2 *Description*

3 Releases the unmanaged resources used by the  
4 **System.Web.UI.Design.WebControls.RegexEditorDialog** and optionally  
5 releases the managed resources.

6 This method is called by the public **Dispose()** method and the  
7 **System.Object.Finalize** method.

8 **lstStandardExpressions\_SelectedIndexChanged**

9  
10 [C#] protected void lstStandardExpressions\_SelectedIndexChanged(object sender,  
11 EventArgs e);

12 [C++] protected: void lstStandardExpressions\_SelectedIndexChanged(Object\*  
13 sender, EventArgs\* e);

14 [VB] Protected Sub lstStandardExpressions\_SelectedIndexChanged(ByVal sender  
15 As Object, ByVal e As EventArgs)

16 [JScript] protected function

17 lstStandardExpressions\_SelectedIndexChanged(sender : Object, e : EventArgs);  
18

19 *Description*

20 Represents the method that will handle the XXX event of a XXX. The  
21 source of the event. An EventArgs that provides data for the event.

22 **RegexTypeEditor\_Activated**

23  
24 [C#] protected void RegexTypeEditor\_Activated(object sender, EventArgs e);

25 [C++] protected: void RegexTypeEditor\_Activated(Object\* sender, EventArgs\*

1 e);

2 [VB] Protected Sub RegexTypeEditor\_Activated(ByVal sender As Object, ByVal  
3 e As EventArgs)

4 [JScript] protected function RegexTypeEditor\_Activated(sender : Object, e :  
5 EventArgs);

6  
7 *Description*

8 Represents the method that will handle the XXX event of a XXX. The  
9 source of the event. An EventArgs that provides data for the event.

10 txtExpression\_TextChanged

11  
12 [C#] protected void txtExpression\_TextChanged(object sender, EventArgs e);

13 [C++] protected: void txtExpression\_TextChanged(Object\* sender, EventArgs\*  
14 e);

15 [VB] Protected Sub txtExpression\_TextChanged(ByVal sender As Object, ByVal  
16 e As EventArgs)

17 [JScript] protected function txtExpression\_TextChanged(sender : Object, e :  
18 EventArgs);

19  
20 *Description*

21 Represents the method that will handle the XXX event of a XXX. The  
22 source of the event. An EventArgs that provides data for the event.

23 RegexTypeEditor class (System.Web.UI.Design.WebControls)

24 WndProc



1  
2  
3 *Description*

4 Provides a user interface for editing regular expressions.

5 RegexTypeEditor

6 *Example Syntax:*

7 WndProc

8  
9 [C#] public RegexTypeEditor();

10 [C++] public: RegexTypeEditor();

11 [VB] Public Sub New()

12 [JScript] public function RegexTypeEditor();

13 EditValue

14  
15 [C#] public override object EditValue(ITypeDescriptorContext context,

16 IServiceProvider provider, object value);

17 [C++] public: Object\* EditValue(ITypeDescriptorContext\* context,

18 IServiceProvider\* provider, Object\* value);

19 [VB] Overrides Public Function EditValue(ByVal context As

20 ITypeDescriptorContext, ByVal provider As IServiceProvider, ByVal value As

21 Object) As Object

22 [JScript] public override function EditValue(context : ITypeDescriptorContext,

23 provider : IServiceProvider, value : Object) : Object;

24  
25 *Description*

Edits the value of the specified object using the specified service provider and context.

*Return Value:* The new value of the object. If the value of the object hasn't changed, this should return the same object it was passed. An **System.ComponentModel.ITypeDescriptorContext** that can provide additional context information. A service provider. The object to edit the value of.

#### GetEditStyle

[C#] public override UITypeEditorEditStyle

GetEditStyle(ITypeDescriptorContext context);

[C++] public: UITypeEditorEditStyle GetEditStyle(ITypeDescriptorContext\* context);

[VB] Overrides Public Function GetEditStyle(ByVal context As ITypeDescriptorContext) As UITypeEditorEditStyle

[JScript] public override function GetEditStyle(context : ITypeDescriptorContext) : UITypeEditorEditStyle;

#### *Description*

Gets the editor style used by the **System.Web.UI.Design.WebControls.RegexTypeEditor.EditValue(System.ComponentModel.ITypeDescriptorContext, System.IServiceProvider, System.Object)** method.

*Return Value:* A **System.Drawing.Design.UITypeEditorEditStyle** that indicates the editor style. An **System.ComponentModel.ITypeDescriptorContext** that may be used to gain additional context information.

1 RepeaterDesigner class (System.Web.UI.Design.WebControls)

2 ToString

3

4

5 *Description*

6 Provides a designer for the **System.Web.UI.WebControls.Repeater**  
7 control.

8 RepeaterDesigner

9 *Example Syntax:*

10 ToString

11

12 [C#] public RepeaterDesigner();

13 [C++] public: RepeaterDesigner();

14 [VB] Public Sub New()

15 [JScript] public function RepeaterDesigner();

16

17 *Description*

18 Initializes a new instance of the

19 **System.Web.UI.Design.WebControls.RepeaterDesigner** class.

20 AllowResize

21 AssociatedComponents

22 Behavior

23 Component

24 DataBindings

25 DataMember

ToString

*Description*

DataSource

ToString

[C#] public string DataSource {get; set;}

[C++] public: \_\_property String\* get\_DataSource();public: \_\_property void  
set\_DataSource(String\*);

[VB] Public Property DataSource As String

[JScript] public function get DataSource() : String;public function set  
DataSource(String);

*Description*

Designer implementation of DataSource property that operates on the  
DataSource property in the control's binding collection.

DesignTimeElement

DesignTimeElementView

DesignTimeHtmlRequiresLoadComplete

ID

InheritanceAttribute

Inherited

IsDirty

1       ReadOnly  
2       ShadowProperties  
3       ShouldCodeSerialize  
4       TemplatesExist  
5       ToString

6  
7  
8       *Description*

9           Verbs  
10          Dispose

11  
12       [C#] protected override void Dispose(bool disposing);  
13       [C++] protected: void Dispose(bool disposing);  
14       [VB] Overrides Protected Sub Dispose(ByVal disposing As Boolean)  
15       [JScript] protected override function Dispose(disposing : Boolean);

16  
17       *Description*

18           Performs the cleanup of the designer class.

19          GetDesignTimeDataSource

20  
21       [C#] protected IEnumerable GetDesignTimeDataSource(int minimumRows);  
22       [C++] protected: IEnumerable\* GetDesignTimeDataSource(int minimumRows);  
23       [VB] Protected Function GetDesignTimeDataSource(ByVal minimumRows As  
24       Integer) As IEnumerable  
25       [JScript] protected function GetDesignTimeDataSource(minimumRows : int) :

1 IEnumerable;

3 *Description*

4 Returns a sample data matching the schema of the selected datasource.

5 *Return Value:* A live datasource for use at design-time. The minimumn rows of  
6 sample data the datasource should contain.

7 GetDesignTimeDataSource

9 [C#] protected IEnumerable GetDesignTimeDataSource(IEnumerable  
10 selectedDataSource, int minimumRows);

11 [C++] protected: IEnumerable\* GetDesignTimeDataSource(IEnumerable\*  
12 selectedDataSource, int minimumRows);

13 [VB] Protected Function GetDesignTimeDataSource(ByVal selectedDataSource  
14 As IEnumerable, ByVal minimumRows As Integer) As IEnumerable

15 [JScript] protected function GetDesignTimeDataSource(selectedDataSource :  
16 IEnumerable, minimumRows : int) : IEnumerable;

18 *Description*

19 Returns a sample data matching the schema of the selected datasource.

20 *Return Value:* A live datasource for use at design-time. The selected datasource to  
21 be used as a reference for the shape of the data. The minimumn rows of sample  
22 data the datasource should contain.

23 GetDesignTimeHtml

25 [C#] public override string GetDesignTimeHtml();

1 [C++] public: String\* GetDesignTimeHtml();

2 [VB] Overrides Public Function GetDesignTimeHtml() As String

3 [JScript] public override function GetDesignTimeHtml() : String;

4  
5 *Description*

6       Retrieves the HTML to be used for the design-time representation of the  
7 control.

8 *Return Value:* Design Time HTML.

9       GetEmptyDesignTimeHtml

10  
11 [C#] protected override string GetEmptyDesignTimeHtml();

12 [C++] protected: String\* GetEmptyDesignTimeHtml();

13 [VB] Overrides Protected Function GetEmptyDesignTimeHtml() As String

14 [JScript] protected override function GetEmptyDesignTimeHtml() : String;

15  
16 *Description*

17       GetErrorDesignTimeHtml

18  
19 [C#] protected override string GetErrorDesignTimeHtml(Exception e);

20 [C++] protected: String\* GetErrorDesignTimeHtml(Exception\* e);

21 [VB] Overrides Protected Function GetErrorDesignTimeHtml(ByVal e As  
22 Exception) As String

23 [JScript] protected override function GetErrorDesignTimeHtml(e : Exception) :  
24 String;

1  
2 *Description*

3       GetResolvedSelectedDataSource

4  
5 [C#] public IEnumerable GetResolvedSelectedDataSource();

6 [C++] public: \_\_sealed IEnumerable\* GetResolvedSelectedDataSource();

7 [VB] NotOverridable Public Function GetResolvedSelectedDataSource() As  
8 IEnumerable

9 [JScript] public function GetResolvedSelectedDataSource() : IEnumerable;

10  
11 *Description*

12       GetSelectedDataSource

13  
14 [C#] public object GetSelectedDataSource();

15 [C++] public: \_\_sealed Object\* GetSelectedDataSource();

16 [VB] NotOverridable Public Function GetSelectedDataSource() As Object

17 [JScript] public function GetSelectedDataSource() : Object;

18  
19 *Description*

20       Retrieves the selected datasource component from the component's  
21 container.

22 *Return Value:* The selected datasource; null if a datasource is not found, or a  
23 datasource with the same name does not exist.

24       Initialize



1  
2 [C#] public override void Initialize(IComponent component);

3 [C++] public: void Initialize(IComponent\* component);

4 [VB] Overrides Public Sub Initialize(ByVal component As IComponent)

5 [JScript] public override function Initialize(component : IComponent);

6  
7 *Description*

8        Initializes the designer with the Repeater control that this instance of the  
9 designer is associated with. The associated Repeater control.

10        OnComponentChanged

11  
12 [C#] public override void OnComponentChanged(object source,

13 ComponentChangedEventArgs ce);

14 [C++] public: void OnComponentChanged(Object\* source,

15 ComponentChangedEventArgs\* ce);

16 [VB] Overrides Public Sub OnComponentChanged(ByVal source As Object,

17 ByVal ce As ComponentChangedEventArgs)

18 [JScript] public override function OnComponentChanged(source : Object, ce :

19 ComponentChangedEventArgs);

20  
21 *Description*

22        Handles changes made to the component. This includes changes made in  
23 the Properties window.

24        OnDataSourceChanged

```

1
2 [C#] public virtual void OnDataSourceChanged();
3 [C++] public: virtual void OnDataSourceChanged();
4 [VB] Overridable Public Sub OnDataSourceChanged()
5 [JScript] public function OnDataSourceChanged();
6

```

#### *Description*

Handles changes made to the data source Handles changes made to the data source

#### **PreFilterProperties**

```

12 [C#] protected override void PreFilterProperties(IDictionary properties);
13 [C++] protected: void PreFilterProperties(IDictionary* properties);
14 [VB] Overrides Protected Sub PreFilterProperties(ByVal properties As
15 IDictionary)
16 [JScript] protected override function PreFilterProperties(properties : IDictionary);
17

```

#### *Description*

Filter the properties to replace the runtime DataSource property descriptor with the designer's.

TableCellsCollectionEditor class (System.Web.UI.Design.WebControls)

UpdateDesignTimeHtml

#### *Description*

Provides a user interface for editing cells in a table.

TableCellsCollectionEditor

*Example Syntax:*

UpdateDesignTimeHtml

[C#] public TableCellsCollectionEditor(Type type);

[C++] public: TableCellsCollectionEditor(Type\* type);

[VB] Public Sub New(ByVal type As Type)

[JScript] public function TableCellsCollectionEditor(type : Type);

### *Description*

Initializes a new instance of the **System.Web.UI.Design.WebControls.TableCellsCollectionEditor** class. The type of the collection to edit.

CollectionItemType

CollectionType

Context

HelpTopic

NewItemTypes

CanSelectMultipleInstances

[C#] protected override bool CanSelectMultipleInstances();

[C++] protected: bool CanSelectMultipleInstances();

[VB] Overrides Protected Function CanSelectMultipleInstances() As Boolean

[JScript] protected override function CanSelectMultipleInstances() : Boolean;

1  
2 *Description*

3 Indicates whether multiple table cells can be selected at the same time.

4 *Return Value:* **true** if multiple cells can be selected at the same time; otherwise,  
5 **false** .

6 *CreateInstance*

7  
8 [C#] protected override object CreateInstance(Type itemType);

9 [C++] protected: Object\* CreateInstance(Type\* itemType);

10 [VB] Overrides Protected Function CreateInstance(ByVal itemType As Type) As  
11 Object

12 [JScript] protected override function CreateInstance(itemType : Type) : Object;

13  
14 *Description*

15 Creates an instance of the editor for use with the specified type.

16 *Return Value:* An object of the specified type. The type to create an instance of.

17 TableDesigner class (System.Web.UI.Design.WebControls)

18 ToString

19  
20  
21 *Description*

22 Provides design-time support for the **System.Web.UI.WebControls.Table**  
23 Web server control.

24 TableDesigner

25 *Example Syntax:*

1	ToString
2	
3	[C#] public TableDesigner();
4	[C++] public: TableDesigner();
5	[VB] Public Sub New()
6	[JScript] public function TableDesigner();
7	AllowResize
8	AssociatedComponents
9	Behavior
10	Component
11	DataBindings
12	DesignTimeElement
13	DesignTimeElementView
14	DesignTimeHtmlRequiresLoadComplete
15	ID
16	InheritanceAttribute
17	Inherited
18	IsDirty
19	ReadOnly
20	ShadowProperties
21	ShouldCodeSerialize
22	Verbs
23	GetDesignTimeHtml
24	
25	[C#] public override string GetDesignTimeHtml();

1 [C++] public: String\* GetDesignTimeHtml();  
 2 [VB] Overrides Public Function GetDesignTimeHtml() As String  
 3 [JScript] public override function GetDesignTimeHtml() : String;

4  
 5 *Description*

6 Gets the HTML that is used to represent the control at design time.

7 *Return Value:* The HTML that is used to represent the control at design time.

8 TableRowsCollectionEditor class (System.Web.UI.Design.WebControls)

9 UpdateDesignTimeHtml

10  
 11  
 12 *Description*

13 Provides a user interface for editing rows of a table.

14 TableRowsCollectionEditor

15 *Example Syntax:*

16 UpdateDesignTimeHtml

17  
 18 [C#] public TableRowsCollectionEditor(Type type);  
 19 [C++] public: TableRowsCollectionEditor(Type\* type);  
 20 [VB] Public Sub New(ByVal type As Type)  
 21 [JScript] public function TableRowsCollectionEditor(type : Type);  
 22

23 *Description*  
 24  
 25

1        Initializes a new instance of the  
2        **System.Web.UI.Design.WebControls.TableRowsCollectionEditor** class. The  
3        type of the collection to edit.

4        **CollectionItemType**

5        **CollectionType**

6        **Context**

7        **HelpTopic**

8        **NewItemTypes**

9        **CanSelectMultipleInstances**

10  
11        [C#] protected override bool CanSelectMultipleInstances();

12        [C++] protected: bool CanSelectMultipleInstances();

13        [VB] Overrides Protected Function CanSelectMultipleInstances() As Boolean

14        [JScript] protected override function CanSelectMultipleInstances() : Boolean;

15  
16        *Description*

17        Indicates whether multiple instances may be selected.

18        *Return Value:* **true** if multiple items can be selected at once; otherwise, **false** .

19        This implementation always returns **false** .

20        **CreateInstance**

21  
22        [C#] protected override object CreateInstance(Type itemType);

23        [C++] protected: Object\* CreateInstance(Type\* itemType);

24        [VB] Overrides Protected Function CreateInstance(ByVal itemType As Type) As

25        Object

1 [JScript] protected override function CreateInstance(itemType : Type) : Object;

2  
3 *Description*

4 Creates an instance of the specified type.

5 *Return Value:* An object of the specified type. The type to create an instance of.

6 XmlDesigner class (System.Web.UI.Design.WebControls)

7 ToString

8  
9  
10 *Description*

11 Provides a designer for the **System.Web.UI.WebControls.Xml Web**  
12 **server control.**

13 XmlDesigner

14 *Example Syntax:*

15 ToString

16  
17 [C#] public XmlDesigner();

18 [C++] public: XmlDesigner();

19 [VB] Public Sub New()

20 [JScript] public function XmlDesigner();

21  
22 *Description*

23 Initializes a new instance of the

24 **System.Web.UI.Design.WebControls.XmlDesigner** class.

25 AllowResize



1	AssociatedComponents
2	Behavior
3	Component
4	DataBindings
5	DesignTimeElement
6	DesignTimeElementView
7	DesignTimeHtmlRequiresLoadComplete
8	ID
9	InheritanceAttribute
10	Inherited
11	IsDirty
12	ReadOnly
13	ShadowProperties
14	ShouldCodeSerialize
15	Verbs
16	Dispose
17	
18	[C#] protected override void Dispose(bool disposing);
19	[C++] protected: void Dispose(bool disposing);
20	[VB] Overrides Protected Sub Dispose(ByVal disposing As Boolean)
21	[JScript] protected override function Dispose(disposing : Boolean);
22	
23	<i>Description</i>
24	Performs the cleanup of the designer class.
25	GetDesignTimeHtml

1  
2 [C#] public override string GetDesignTimeHtml();

3 [C++] public: String\* GetDesignTimeHtml();

4 [VB] Overrides Public Function GetDesignTimeHtml() As String

5 [JScript] public override function GetDesignTimeHtml() : String;

6  
7 *Description*

8 Gets the HTML that is used to represent the control at design time.

9 *Return Value:* The HTML that is used to represent the control at design time.

10 GetEmptyDesignTimeHtml

11  
12 [C#] protected override string GetEmptyDesignTimeHtml();

13 [C++] protected: String\* GetEmptyDesignTimeHtml();

14 [VB] Overrides Protected Function GetEmptyDesignTimeHtml() As String

15 [JScript] protected override function GetEmptyDesignTimeHtml() : String;

16  
17 *Description*

18 Gets the HTML that is used to fill an empty control.

19 *Return Value:* The HTML used to fill an empty control.

20 Initialize

21  
22 [C#] public

## System.Web.UI.HtmlControls

### *Description*

The **System.Web.UI.HtmlControls** namespace is a collection of classes that allow you to create HTML server controls on a Web page. HTML server controls run on the server and map directly to standard HTML tags supported by all browsers. This allows you to programmatically control the HTML elements on the Web page.

HtmlAnchor class (**System.Web.UI.HtmlControls**)

### *Description*

Defines the methods, properties, and events for the **System.Web.UI.HtmlControls.HtmlAnchor** control. This class allows programmatic access to the HTML tag on the server.

There are two ways to use the **System.Web.UI.HtmlControls.HtmlAnchor** class. The first is for navigation: using the **System.Web.UI.HtmlControls.HtmlAnchor.HRef** property to define the location of the page to link to. The second is for postback events: using the **System.Web.UI.HtmlControls.HtmlAnchor.ServerClick** event to programmatically handle the user's click on a link.

Constructors:

HtmlAnchor

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25

3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

24  
25

1 Gets or sets the URL target of the link specified in the  
2 **System.Web.UI.HtmlControls.HtmlAnchor** server control.

3 Use this property to specify the URL to link to when the  
4 **System.Web.UI.HtmlControls.HtmlAnchor** is clicked.

5 ID

6 InnerHtml

7 InnerText

8 IsTrackingViewState

9 Name

10  
11  
12 **Description**

13 Gets or sets the bookmark name defined in the  
14 **System.Web.UI.HtmlControls.HtmlAnchor** server control.

15 Use this property to mark sections on a Web page with a name. This  
16 allows you to link to this section from anywhere on the same page. For  
17 example, you can provide a table of contents at the top of a page that will link  
18 directly to topics on the page.

19 NamingContainer

20 Page

21 Parent

22 Site

23 Style

24 TagName

25 Target

1  
2  
3 **Description**

4 Gets or sets the target window or frame to load Web page content into.

5 Use this property to specify the frame or window that displays the Web  
6 page linked to.

7 TemplateSourceDirectory

8 Title

9  
10  
11 **Description**

12 Gets or sets the title that the browser displays for a Web page.

13 Use this property to specify a custom title when the browser identifies  
14 the page you link to. This property also may be used to provide a custom tool  
15 tip for the System.Web.UI.HtmlControls.HtmlAnchor control.

16 UniqueID

17 ViewState

18 ViewStateIgnoresCase

19 Visible

20  
21  
22 **Description**

23 Occurs on the server when a user clicks the  
24 System.Web.UI.HtmlControls.HtmlAnchor control on the browser.

1           This           event           is           raised           when           the  
2   System.Web.UI.HtmlControls.HtmlAnchor is clicked.

3           Methods:

4           OnServerClick

6   [C#]   protected   virtual   void   OnServerClick(EventArgs   e);

7   [C++]   protected:   virtual   void   OnServerClick(EventArgs\*   e);

8   [VB]   Overridable Protected Sub OnServerClick(ByVal e As EventArgs)

9   [JScript]   protected   function   OnServerClick(e   :   EventArgs);

11   ***Description***

12           Raises   the   System.Web.UI.HtmlControls.HtmlAnchor.ServerClick  
13   event. This allows you to handle the event directly.

14           This server event causes a roundtrip to occur from the client to the  
15   server and back. An System.EventArgs that contains event data.

16           RenderAttributes

18   [C#]   protected   override   void   RenderAttributes(HtmlTextWriter   writer);

19   [C++]   protected:   void   RenderAttributes(HtmlTextWriter\*   writer);

20   [VB]   Overrides   Protected   Sub   RenderAttributes(ByVal   writer   As  
21   HtmlTextWriter)

22   [JScript]   protected   override   function   RenderAttributes(writer   :  
23   HtmlTextWriter);

25   ***Description***

1 IPostBackEventHandler.RaisePostBackEvent

2

3 [C#] void IPostBackEventHandler.RaisePostBackEvent(string  
4 eventArgument);

5 [C++] void IPostBackEventHandler::RaisePostBackEvent(String\*  
6 eventArgument);

7 [VB] Sub RaisePostBackEvent(ByVal eventArgument As String) Implements  
8 IPostBackEventHandler.RaisePostBackEvent

9 [JScript] function  
10 IPostBackEventHandler.RaisePostBackEvent(eventArgument : String);

11 HtmlButton class (System.Web.UI.HtmlControls)  
12 ViewState

13

14

15 **Description**

16 Defines the methods, properties, and events for the  
17 System.Web.UI.HtmlControls.HtmlButton control. This class allows  
18 programmatic access to the HTML tag on the server.

19 The element allows Web developers to create UI form buttons that can  
20 be composed of embedded HTML elements, including other server controls.

21 HtmlButton

22 **Example Syntax:**  
23 ViewState

24

25 [C#] public HtmlButton();



1	[C++]	public:	HtmlButton();
2	[VB]	Public	Sub New()
3	[JScript]	public	function HtmlButton();

**Description**

Initializes a new instance of the System.Web.UI.HtmlControls.HtmlButton class.

Attributes

CausesValidation

TrackViewState

**Description**

Gets or sets a value indicating whether validation is performed when the System.Web.UI.HtmlControls.HtmlButton control is clicked.

By default, page validation is performed when an System.Web.UI.HtmlControls.HtmlButton control is clicked. Page validation determines whether the input controls associated with a validation control on the page all pass the validation rules specified by the validation control.

ChildControlsCreated

ClientID

Context

Controls

Disabled

EnableViewState

1	Events
2	HasChildViewState
3	ID
4	InnerHTML
5	InnerText
6	IsTrackingViewState
7	NamingContainer
8	Page
9	Parent
10	Site
11	Style
12	TagName
13	TemplateSourceDirectory
14	UniqueID
15	ViewState
16	ViewStateIgnoresCase
17	Visible
18	TrackViewState

19

20

21 **Description**

22       **Occurs**       **when**       **the**       **user**       **clicks**       **an**

23 **System.Web.UI.HtmlControls.HtmlButton** control on the client Web page.

24       **This**       **event**       **is**       **raised**       **when**       **the**

25 **System.Web.UI.HtmlControls.HtmlButton** is clicked.

## OnServerClick

```
[C#]    protected    virtual    void    OnServerClick(EventArgs    e);  
[C++]    protected:    virtual    void    OnServerClick(EventArgs*    e);  
[VB]    Overridable Protected Sub OnServerClick(ByVal e As EventArgs)  
[JScript]    protected    function    OnServerClick(e    :    EventArgs);
```

### *Description*

Raises the System.Web.UI.HtmlControls.HtmlButton.ServerClick event. This allows you to handle the event directly.

This event causes a roundtrip to occur from the client to the server and back. It is deliberately different from the client-side OnClick event. In the event that a conflict exists between code run with a System.Web.UI.HtmlControls.HtmlButton.ServerClick event and code run by a client-side OnClick event, the server-side event instructions will override the client-side code. An System.EventArgs that contains the event data.

## RenderAttributes

```
[C#]    protected override void RenderAttributes(HtmlTextWriter writer);  
[C++]    protected:    void    RenderAttributes(HtmlTextWriter*    writer);  
[VB]    Overrides Protected Sub RenderAttributes(ByVal writer As  
HtmlTextWriter)  
[JScript]    protected    override    function    RenderAttributes(writer    :  
HtmlTextWriter);
```

**Description**

**IPostBackEventHandler.RaisePostBackEvent**

**[C#] void IPostBackEventHandler.RaisePostBackEvent(string eventArgument);**

**[C++] void IPostBackEventHandler::RaisePostBackEvent(String\* eventArgument);**

**[VB] Sub RaisePostBackEvent(ByVal eventArgument As String) Implements IPostBackEventHandler.RaisePostBackEvent**

**[JScript] function**

**IPostBackEventHandler.RaisePostBackEvent(eventArgument : String);**

HtmlContainerControl class (System.Web.UI.HtmlControls)

TrackViewState

**Description**

**Defines the methods, properties, and events available to all HTML server controls that must have a closing tag.**

**The most common controls with a closing tag are the**  
**Top of Form**

, ,

, , and elements.

HtmlContainerControl

*Example Syntax:*

TrackViewState

```
[C#]                public                HtmlContainerControl();
[C++]                public:                HtmlContainerControl();
[VB]                Public                Sub                New()
[JScript] public function HtmlContainerControl(); Initializes a new instance of the
System.Web.UI.HtmlControls.HtmlContainerControl class.
```

*Description*

Initializes a new instance of the **System.Web.UI.HtmlControls.HtmlContainerControl** class using default values.

Use this constructor to create and initialize a new instance of the **System.Web.UI.HtmlControls.HtmlContainerControl** class using default values.

HtmlContainerControl

*Example Syntax:*

TrackViewState

```
[C#]                public                HtmlContainerControl(string                tag);
[C++]                public:                HtmlContainerControl(String*                tag);
[VB]                Public                Sub                New(ByVal                tag                As                String)
[JScript] public function HtmlContainerControl(tag                :                String);
```

1  
2 *Description*

3       Initializes       a       new       instance       of       the  
4       **System.Web.UI.HtmlControls.HtmlContainerControl** class using the specified  
5       tag name.

6       Use this constructor to create and initialize a new instance of the  
7       **System.Web.UI.HtmlControls.HtmlContainerControl** class using the specified  
8       tag. A string that specifies the tag name of the control.

9       Attributes

10      ChildControlsCreated

11      ClientID

12      Context

13      Controls

14      Disabled

15      EnableViewState

16      Events

17      HasChildViewState

18      ID

19      InnerHtml

20      TrackViewState

21  
22  
23 *Description*

24       Gets or sets the content found between the opening and closing tags of the  
25       specified HTML server control.

the

InnerText

```
[C#] public virtual string InnerText {get; set;}
```

```
[C++] public: __property virtual String* get_InnerText();public: __property
```

```
virtual                                void                                set_InnerText(String*);
```

```
[VB]      Overridable      Public      Property      InnerText      As      String
```

```
[JScript] public function get InnerText() : String;public function set
```

```
InnerText(String);
```

Gets or sets the text between the opening and closing tags of the specified HTML server control.

the

**System.Web.UI.HtmlControls.HtmlContainerControl.InnerText** property to programmatically modify the contents between the opening and closing tags of an HTML server control.

## IsTrackingViewState

## NamingContainer

Page

Parent

1 Site  
2 Style  
3 TagName  
4 TemplateSourceDirectory  
5 UniqueID  
6 ViewState  
7 ViewStateIgnoresCase  
8 Visible  
9 CreateControlCollection

10

11 [C#] protected override ControlCollection CreateControlCollection();  
12 [C++] protected: ControlCollection\* CreateControlCollection();  
13 [VB] Overrides Protected Function CreateControlCollection() As  
14 ControlCollection  
15 [JScript] protected override function CreateControlCollection() :  
16 ControlCollection;  
17 LoadViewState

18

19 [C#] protected override void LoadViewState(object savedState);  
20 [C++] protected: void LoadViewState(Object\* savedState);  
21 [VB] Overrides Protected Sub LoadViewState(ByVal savedState As Object)  
22 [JScript] protected override function LoadViewState(savedState : Object);

23

24 *Description*

25 Render



```

1
2 [C#]    protected    override    void    Render(HtmlTextWriter    writer);
3 [C++]    protected:    void    Render(HtmlTextWriter*    writer);
4 [VB]    Overrides    Protected    Sub    Render(ByVal    writer    As    HtmlTextWriter)
5 [JScript]    protected    override    function    Render(writer    :    HtmlTextWriter);
6

```

### *Description*

#### RenderAttributes

```

10 [C#]    protected    override    void    RenderAttributes(HtmlTextWriter    writer);
11 [C++]    protected:    void    RenderAttributes(HtmlTextWriter*    writer);
12 [VB]    Overrides    Protected    Sub    RenderAttributes(ByVal    writer    As
13 HtmlTextWriter)
14 [JScript]    protected    override    function    RenderAttributes(writer    :    HtmlTextWriter);
15

```

### *Description*

#### RenderEndTag

```

19 [C#]    protected    virtual    void    RenderEndTag(HtmlTextWriter    writer);
20 [C++]    protected:    virtual    void    RenderEndTag(HtmlTextWriter*    writer);
21 [VB]    Overridable    Protected    Sub    RenderEndTag(ByVal    writer    As
22 HtmlTextWriter)
23 [JScript]    protected    function    RenderEndTag(writer    :    HtmlTextWriter);
24

```

### *Description*

1       HtmlControl class (System.Web.UI.HtmlControls)

2       TrackViewState

3  
4  
5       *Description*

6       Defines the methods, properties, and events common to all HTML server  
7       controls in the Web Forms page framework.

8       The **System.Web.UI.HtmlControls.HtmlControl** class provides common  
9       properties inherited by all HTML server control classes. It is not instantiated  
10      directly.

11      HtmlControl

12      *Example Syntax:*

13      TrackViewState

14  
15      [C#]                               public                               HtmlControl();

16      [C++]                              public:                             HtmlControl();

17      [VB]                              Public                             Sub                             New()

18      [JavaScript] public function HtmlControl(); Initializes a new instance of the

19      **System.Web.UI.HtmlControls.HtmlControl** class.

20  
21      *Description*

22      Initializes a new instance of the  
23      **System.Web.UI.HtmlControls.HtmlControl** class using default values.

24      This constructor is used to create and initialize a new instance of the  
25      **System.Web.UI.HtmlControls.HtmlControl** class using default values.

```

1      HtmlControl
2      Example Syntax:
3      ViewState
4
5      [C#]          public          HtmlControl(string          tag);
6      [C++]          public:          HtmlControl(String*          tag);
7      [VB]      Public      Sub      New(ByVal      tag      As      String)
8      [JScript]      public      function      HtmlControl(tag      :      String);
9

```

#### Description

Initializes a new instance of the **System.Web.UI.HtmlControls.HtmlControl** class using the specified tag.

This constructor is used to create and initialize a new instance of the **System.Web.UI.HtmlControls.HtmlControl** class using the specified tag. A string that specifies the tag name of the control.

#### Attributes

ViewState

```

19      [C#]          public          AttributeCollection          Attributes          {get;}
20      [C++]          public:          __property          AttributeCollection*          get_Attributes();
21      [VB]      Public      ReadOnly      Property      Attributes      As      AttributeCollection
22      [JScript]      public      function      get      Attributes()      :      AttributeCollection;
23

```

#### Description

1 Gets a collection of all attribute name and value pairs expressed on a server  
2 control tag within the .aspx file.

3 Use this property to programmatically access the attributes of the HTML  
4 server control. All Html server controls store their attributes in the  
5 **System.Web.UI.Control.ViewState** .

6 ChildControlsCreated

7 ClientID

8 Context

9 Controls

10 Disabled

11 TrackViewState

12  
13  
14 *Description*

15 Gets or sets a value indicating whether the HTML server control is  
16 disabled.

17 On the browser, a disabled element or control is read-only, with the  
18 following added restrictions: its value is not submitted with the form, the element  
19 or control cannot receive focus, and the element or control is skipped when  
20 navigating the document by tabbing.

21 EnableViewState

22 Events

23 HasChildViewState

24 ID

25 IsTrackingViewState

1       NamingContainer  
 2       Page  
 3       Parent  
 4       Site  
 5       Style  
 6       TrackViewState

9       *Description*

10       Gets a collection of all cascading style sheet (CSS) properties applied to a  
 11       specified HTML server control in the .aspx file.

12       Use this property to programmatically access the style properties of the  
 13       HTML server control.

14       TagName  
 15       TrackViewState

16  
 17       [C#]       public       virtual       string       TagName       {get;}  
 18       [C++]       public:       \_\_property       virtual       String\*       get\_TagName();  
 19       [VB]       Overridable   Public   ReadOnly   Property   TagName   As   String  
 20       [JScript]       public       function       get       TagName()       :       String;

22       *Description*

23       Gets the element name of a tag that contains a **runat=server** attribute and  
 24       value pair.

1 Use this property to programmatically determine the element name of the  
2 HTML server control.

3 TemplateSourceDirectory

4 UniqueID

5 ViewState

6 ViewStateIgnoresCase

7 TrackViewState

8  
9  
10 *Description*

11 Visible

12 CreateControlCollection

13  
14 [C#] protected override ControlCollection CreateControlCollection();

15 [C++] protected: ControlCollection\* CreateControlCollection();

16 [VB] Overrides Protected Function CreateControlCollection() As  
17 ControlCollection

18 [JScript] protected override function CreateControlCollection() :  
19 ControlCollection;

20 Render

21  
22 [C#] protected override void Render(HtmlTextWriter writer);

23 [C++] protected: void Render(HtmlTextWriter\* writer);

24 [VB] Overrides Protected Sub Render(ByVal writer As HtmlTextWriter)

25 [JScript] protected override function Render(writer : HtmlTextWriter);

1  
2 *Description*

3       RenderAttributes

4  
5 [C#]   protected   virtual   void   RenderAttributes(HtmlTextWriter   writer);  
6 [C++]   protected:   virtual   void   RenderAttributes(HtmlTextWriter\*   writer);  
7 [VB]   Overridable   Protected   Sub   RenderAttributes(ByVal   writer   As  
8   HtmlTextWriter)  
9 [JScript]   protected   function   RenderAttributes(writer   :   HtmlTextWriter);

10  
11 *Description*

12       RenderBeginTag

13  
14 [C#]   protected   virtual   void   RenderBeginTag(HtmlTextWriter   writer);  
15 [C++]   protected:   virtual   void   RenderBeginTag(HtmlTextWriter\*   writer);  
16 [VB]   Overridable   Protected   Sub   RenderBeginTag(ByVal   writer   As  
17   HtmlTextWriter)  
18 [JScript]   protected   function   RenderBeginTag(writer   :   HtmlTextWriter);

19  
20 *Description*

21       IAttributeAccessor.GetAttribute

22  
23 [C#]       string       IAttributeAccessor.GetAttribute(string       name);  
24 [C++]       String\*       IAttributeAccessor::GetAttribute(String\*       name);  
25 [VB]   Function   GetAttribute(ByVal   name   As   String)   As   String   Implements

1 IAttributeAccessor.GetAttribute

2 [JScript] function IAttributeAccessor.GetAttribute(name : String) : String;

3 IAttributeAccessor.SetAttribute

4  
5 [C#] void IAttributeAccessor.SetAttribute(string name, string value);

6 [C++] void IAttributeAccessor::SetAttribute(String\* name, String\* value);

7 [VB] Sub SetAttribute(ByVal name As String, ByVal value As String)

8 Implements IAttributeAccessor.SetAttribute

9 [JScript] function IAttributeAccessor.SetAttribute(name : String, value : String);

10 HtmlForm class (System.Web.UI.HtmlControls)

11 TrackViewState

12  
13  
14 *Description*

15 Provides programmatic access to the HTML

16 Top of Form

17 element on the server.

18 The **System.Web.UI.HtmlControls.HtmlForm** control is used as a  
19 container for server controls on a Web page. All server controls that post back to  
20 the server must be placed between the opening and closing tags of an  
21 **System.Web.UI.HtmlControls.HtmlForm** control.

22 HtmlForm

23 *Example Syntax:*

24 TrackViewState

25



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#]                public                HtmlForm();  
[C++]              public:                HtmlForm();  
[VB]                Public                Sub                New()  
[JScript]           public                function          HtmlForm();
```

*Description*

Initializes a new instance of the **System.Web.UI.HtmlControls.HtmlForm** class.

Use this constructor to create and initialize a new instance of the **System.Web.UI.HtmlControls.HtmlForm** class.

- Attributes
- ChildControlsCreated
- ClientID
- Context
- Controls
- Disabled
- EnableViewState
- Enctype
- TrackViewState

*Description*

Gets or sets the encoding type browsers use when posting the form's data to the server.

PDF GENERATED BY PDF ELEMENTS

1 Use this property to specify the encoding type browsers use to post data  
2 back to the server.

3 Events

4 HasChildViewState

5 ID

6 InnerHtml

7 InnerText

8 IsTrackingViewState

9 Method

10 TrackViewState

11  
12  
13 *Description*

14 Gets or sets a value that indicates how a browser posts form data to the  
15 server for processing.

16 Use this property to specify how the browser sends form data to the server  
17 for processing. The two common methods supported by all browsers are **GET** and  
18 **POST**.

19 Name

20 TrackViewState

21  
22 [C#] public virtual string Name {get; set;}

23 [C++] public: \_\_property virtual String\* get\_Name();public: \_\_property virtual  
24 void set\_Name(String\*);

25 [VB] Overridable Public Property Name As String

1 [JScript] public function get Name() : String;public function set Name(String);

2  
3 *Description*

4 Gets the identifier name for the  
5 **System.Web.UI.HtmlControls.HtmlForm** control.

6 Use this property to identify a specific  
7 **System.Web.UI.HtmlControls.HtmlForm** control.

8 NamingContainer

9 Page

10 Parent

11 Site

12 Style

13 TagName

14 Target

15 ViewState

16  
17  
18 *Description*

19 Gets or sets the frame or window to render the results of information posted  
20 to the server.

21 Use this property to display the results of information posted to the server  
22 in another browser window or frame.

23 TemplateSourceDirectory

24 UniqueID

25 ViewState

1 ViewStateIgnoresCase  
 2 Visible  
 3 OnInit  
 4  
 5 [C#] protected override void OnInit(EventArgs e);  
 6 [C++] protected: void OnInit(EventArgs\* e);  
 7 [VB] Overrides Protected Sub OnInit(ByVal e As EventArgs)  
 8 [JScript] protected override function OnInit(e : EventArgs);  
 9

#### 10 *Description*

11 Call RegisterViewStateHandler().

#### 12 Render

13  
 14 [C#] protected override void Render(HtmlTextWriter output);  
 15 [C++] protected: void Render(HtmlTextWriter\* output);  
 16 [VB] Overrides Protected Sub Render(ByVal output As HtmlTextWriter)  
 17 [JScript] protected override function Render(output : HtmlTextWriter);  
 18

#### 19 *Description*

#### 21 RenderAttributes<sup>2</sup>

22  
 23 [C#] protected override void RenderAttributes(HtmlTextWriter writer);  
 24 [C++] protected: void RenderAttributes(HtmlTextWriter\* writer);  
 25 [VB] Overrides Protected Sub RenderAttributes(ByVal writer As

1 HtmlTextWriter)

2 [JScript] protected override function RenderAttributes(writer : HtmlTextWriter);

3  
4 *Description*

5       RenderChildren

6  
7 [C#] protected override void RenderChildren(HtmlTextWriter writer);

8 [C++] protected: void RenderChildren(HtmlTextWriter\* writer);

9 [VB] Overrides Protected Sub RenderChildren(ByVal writer As HtmlTextWriter)

10 [JScript] protected override function RenderChildren(writer : HtmlTextWriter);

11  
12 *Description*

13       HtmlGenericControl class (System.Web.UI.HtmlControls)

14       TrackViewState

15  
16  
17 *Description*

18       Defines the methods, properties, and events for all HTML server control  
19 tags not represented by a specific .NET Framework class.

20       Use this class to represent an HTML server control tag not directly  
21 represented by a .NET Framework class, such as ,

22       ,

23       , and .

24       HtmlGenericControl

25       *Example Syntax:*

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25

### Description

The following table shows initial property values for an instance of **System.Web.UI.HtmlControls.HtmlGenericControl**.

*Example Syntax:*

[C#]	public	HtmlGenericControl(string tag);			
[C++]	public:	HtmlGenericControl(String* tag);			
[VB]	Public	Sub	New(ByVal	tag	As String)
[JScript]	public	function	HtmlGenericControl(tag : String);		

### Description

1        Initializes        a        new        instance        of        the  
2        **System.Web.UI.HtmlControls.HtmlGenericControl** class with the specified  
3        tag.

4        The following table shows initial property values for an instance of  
5        **System.Web.UI.HtmlControls.HtmlGenericControl** . The name of the element  
6        for which the instance of this class is created.

7        Attributes

8        ChildControlsCreated

9        ClientID

10       Context

11       Controls

12       Disabled

13       EnableViewState

14       Events

15       HasChildViewState

16       ID

17       InnerHtml

18       InnerText

19       IsTrackingViewState

20       NamingContainer

21       Page

22       Parent

23       Site

24       Style

25       TagName

TrackViewState

*Description*

Gets or sets the tag name of an element that contains a **runat= "server"** attribute.

Use this property to dynamically change the tag name of a generic control on the Web page.

TemplateSourceDirectory

UniqueID

ViewState


ViewStateIgnoresCase

Visible

HtmlImage class (System.Web.UI.HtmlControls)

TrackViewState

*Description*

Provides programmatic access for the HTML  element on the server.

Use this control to display an image on a Web page. The **System.Web.UI.HtmlControls.HtmlImage** control can be programmatically manipulated to change the image displayed, the image size, and the alignment of the image relative to other page elements.

HtmlImage

*Example Syntax:*



```

1      TrackViewState
2
3  [C#]          public          HtmlImage();
4  [C++]         public:         HtmlImage();
5  [VB]          Public          Sub          New()
6  [JScript]     public          function     HtmlImage();
7

```

8 *Description*

9        Initializes        a        new        instance        of        the

10        **System.Web.UI.HtmlControls.HtmlImage** class.

11        Use this constructor to create and initialize a new instance of the

12        **System.Web.UI.HtmlControls.HtmlImage** class.

13        Align

14        TrackViewState

```

15
16  [C#]          public          string          Align          {get;          set;}
17  [C++]         public:  __property  String*  get_Align();public:  __property  void
18  set_Align(String*);
19  [VB]          Public          Property          Align          As          String
20  [JScript]     public  function  get  Align() : String;public  function  set  Align(String);
21

```

22 *Description*

23        Gets or sets the alignment of the image relative to other Web page

24        elements.

25

1        Use this property to specify the alignment of the image with respect to  
2 other elements on the Web page.

3        Alt

4        TrackViewState

6        [C#]            public            string            Alt            {get;            set;}

7        [C++]    public:    \_\_property    String\*    get\_Alt();public:    \_\_property    void  
8        set\_Alt(String\*);

9        [VB]            Public            Property            Alt            As            String

10       [JScript] public function get Alt() : String;public function set Alt(String);

12       *Description*

13       Gets or sets the alternative caption the browser displays if an image is  
14 unavailable or currently downloading and not yet finished.

15       Use this property to specify the caption displayed when the image specified  
16 by the **System.Web.UI.HtmlControls.HtmlImage.Src** property is unavailable.  
17 On newer browsers, this caption also appears as a ToolTip.

18       Attributes

19       Border

20       TrackViewState

23       *Description*

24       Gets or sets the width of a frame for an image.

1 Use this property to create a frame with the specified width (in pixels) for  
2 an image.

3 ChildControlsCreated

4 ClientID

5 Context

6 Controls

7 Disabled

8 EnableViewState

9 Events

10 HasChildViewState

11 Height

12 TrackViewState

13  
14  
15 *Description*

16 Gets or sets the height of the image.

17 The **System.Web.UI.HtmlControls.HtmlImage.Height** and  
18 **System.Web.UI.HtmlControls.HtmlImage.Width** properties can be used two  
19 ways. You can use the **System.Web.UI.HtmlControls.HtmlImage.Height** and  
20 **System.Web.UI.HtmlControls.HtmlImage.Width** properties to send image size  
21 specifications to the browser. This displays the Web page faster because the  
22 browser does not need to recalculate the positions of elements on the page when  
23 the image loads.

24 ID

25 IsTrackingViewState

1 NamingContainer  
2 Page  
3 Parent  
4 Site  
5 Src  
6 TrackViewState

7  
8  
9 *Description*

10 Gets or sets the source of the image file to display.

11 Use this property to specify the path to the image file to display. If the  
12 image file is in the same directory as the Web page source file that uses it, you can  
13 just specify the file name. Otherwise, you must also include the path to the file.  
14 The path can be absolute or relative to the directory that contains the Web page  
15 source file.

16 Style

17 TagName

18 TemplateSourceDirectory

19 UniqueID

20 ViewState

21 ViewStateIgnoresCase

22 Visible

23 Width

24 TrackViewState  
25

1  
2  
3 *Description*

4 Gets or sets the width of the image.

5 The **System.Web.UI.HtmlControls.HtmlImage.Height** and  
6 **System.Web.UI.HtmlControls.HtmlImage.Width** properties can be used two  
7 ways. You can use the **System.Web.UI.HtmlControls.HtmlImage.Height** and  
8 **System.Web.UI.HtmlControls.HtmlImage.Width** properties to send the image  
9 size specifications to the browser. This displays the Web page faster because the  
10 browser does not need to recalculate the positions of elements on the page when  
11 the image loads.

12 **RenderAttributes**

13  
14 [C#] protected override void RenderAttributes(HtmlTextWriter writer);

15 [C++] protected: void RenderAttributes(HtmlTextWriter\* writer);

16 [VB] Overrides Protected Sub RenderAttributes(ByVal writer As  
17 HtmlTextWriter)



18 [JScript] protected override function RenderAttributes(writer : HtmlTextWriter);

19  
20 *Description*

21 **HtmlInputButton** class (System.Web.UI.HtmlControls)

22 **TrackViewState**

23  
24  
25 *Description*

Allows programmatic access to the HTML ,  , and  elements on the server.

Use this class to create button controls on a Web page.

HtmlInputButton

*Example Syntax:*

TrackViewState

[C#] public HtmlInputButton();

[C++] public: HtmlInputButton();

[VB] Public Sub New()

[JScript] public function HtmlInputButton(); Initializes a new instance of an

**System.Web.UI.HtmlControls.HtmlInputButton** class.

#### Description

Initializes a new instance of an **System.Web.UI.HtmlControls.HtmlInputButton** class using default values.

The following table shows the initial property value for an instance of **System.Web.UI.HtmlControls.HtmlInputButton** .

HtmlInputButton

*Example Syntax:*

TrackViewState

[C#] public HtmlInputButton(string type);

[C++] public: HtmlInputButton(String\* type);

[VB] Public Sub New(ByVal type As String)

1 [JScript] public function HtmlInputButton(type : String);

2  
3 *Description*

4 Initializes a new instance of an  
5 **System.Web.UI.HtmlControls.HtmlInputButton** class using the specified  
6 button type.

7 The following table shows the initial property value for an instance of  
8 **System.Web.UI.HtmlControls.HtmlInputButton** . The input button type.

9 Attributes

10 CausesValidation

11 TrackViewState

12  
13  
14 *Description*

15 Gets or sets a value indicating whether validation is performed when the  
16 **System.Web.UI.HtmlControls.HtmlInputButton** control is clicked.

17 By default, page validation is performed when an  
18 **System.Web.UI.HtmlControls.HtmlInputButton** control is clicked. Page  
19 validation determines whether the input controls associated with a validation  
20 control on the page all pass the validation rules specified by the validation control.

21 ChildControlsCreated

22 ClientID

23 Context

24 Controls

25 Disabled

- 1 EnableViewState
- 2 Events
- 3 HasChildViewState
- 4 ID
- 5 IsTrackingViewState
- 6 Name
- 7 NamingContainer
- 8 Page
- 9 Parent
- 10 Site
- 11 Style
- 12 TagName
- 13 TemplateSourceDirectory
- 14 Type
- 15 UniqueID
- 16 Value
- 17 ViewState
- 18 ViewStateIgnoresCase
- 19 Visible
- 20 TrackViewState

23 *Description*

24 Occurs when an **System.Web.UI.HtmlControls.HtmlInputButton** control  
 25 is clicked on the Web page.



This event is raised when an **System.Web.UI.HtmlControls.HtmlInputButton** control is clicked.

### OnServerClick

```
[C#]    protected    virtual    void    OnServerClick(EventArgs    e);  
[C++]    protected:    virtual    void    OnServerClick(EventArgs*    e);  
[VB]    Overridable Protected Sub OnServerClick(ByVal e As EventArgs)  
[JScript]    protected    function    OnServerClick(e    :    EventArgs);
```

### *Description*

Raises the **System.Web.UI.HtmlControls.HtmlInputButton.ServerClick** event. This allows you to handle the event directly.

The **System.Web.UI.HtmlControls.HtmlInputButton.ServerClick** event is raised when an **System.Web.UI.HtmlControls.HtmlInputButton** control is clicked. A **System.EventArgs** that contains the event data.

### RenderAttributes

```
[C#]    protected    override    void    RenderAttributes(HtmlTextWriter    writer);  
[C++]    protected:    void    RenderAttributes(HtmlTextWriter*    writer);  
[VB]    Overrides Protected Sub RenderAttributes(ByVal    writer    As  
HtmlTextWriter)  
[JScript]    protected    override    function    RenderAttributes(writer : HtmlTextWriter);
```

### *Description*

### IPostBackEventHandler.RaisePostBackEvent

```

1
2 [C#] void IPostBackEventHandler.RaisePostBackEvent(string eventArgument);
3 [C++] void IPostBackEventHandler::RaisePostBackEvent(String*
4 eventArgument);
5 [VB] Sub RaisePostBackEvent(ByVal eventArgument As String) Implements
6 IPostBackEventHandler.RaisePostBackEvent
7 [JScript] function IPostBackEventHandler.RaisePostBackEvent(eventArgument :
8 String);

```

HtmlInputCheckBox class (System.Web.UI.HtmlControls)

TrackViewState

### Description

Allows programmatic access to the HTML ☐ element on the server.

Use the **System.Web.UI.HtmlControls.HtmlInputCheckBox** control to allow the user to select a **true** or **false** state.

HtmlInputCheckBox

### Example Syntax:

TrackViewState

```

21 [C#] public HtmlInputCheckBox();
22 [C++] public: HtmlInputCheckBox();
23 [VB] Public Sub New()
24 [JScript] public function HtmlInputCheckBox();
25

```

1  
2 *Description*

3       Initializes       a       new       instance       of       an  
4 **System.Web.UI.HtmlControls.HtmlInputCheckBox** class.

5       Use this constructor to create and initialize a new instance of the  
6 **System.Web.UI.HtmlControls.HtmlInputCheckBox** class.

7       Attributes

8       Checked

9       TrackViewState

10  
11  
12 *Description*

13       Gets       or       sets       a       value       indicating       whether       the  
14 **System.Web.UI.HtmlControls.HtmlInputCheckBox** is checked.

15       Use       this       property       to       determine       whether       the  
16 **System.Web.UI.HtmlControls.HtmlInputCheckBox** control is checked. This  
17 property can also be used to programmatically set the state of the  
18 **System.Web.UI.HtmlControls.HtmlInputCheckBox** control.

19       ChildControlsCreated

20       ClientID

21       Context

22       Controls

23       Disabled

24       EnableViewState

25       Events

1	HasChildViewState
2	ID
3	IsTrackingViewState
4	Name
5	NamingContainer
6	Page
7	Parent
8	Site
9	Style
10	TagName
11	TemplateSourceDirectory
12	Type
13	UniqueID
14	Value
15	ViewState
16	ViewStateIgnoresCase
17	Visible
18	TrackViewState

21 *Description*

22       Occurs when the Web page is submitted to the server and the

23 **System.Web.UI.HtmlControls.HtmlInputCheckBox** control changes state from

24 the previous post.

25

This event is raised when the Web page is submitted to the server and the state of the **System.Web.UI.HtmlControls.HtmlInputCheckBox** control changes state from the previous post.

#### OnPreRender

```
[C#]      protected      override      void      OnPreRender(EventArgs      e);
[C++]      protected:      void      OnPreRender(EventArgs*      e);
[VB]      Overrides Protected Sub OnPreRender(ByVal e As EventArgs)
[JScript] protected override function OnPreRender(e : EventArgs);
```

#### *Description*

#### OnServerChange

```
[C#]      protected      virtual      void      OnServerChange(EventArgs      e);
[C++]      protected:      virtual      void      OnServerChange(EventArgs*      e);
[VB]      Overridable Protected Sub OnServerChange(ByVal e As EventArgs)
[JScript] protected      function      OnServerChange(e      :      EventArgs);
```

#### *Description*

Raises the **System.Web.UI.HtmlControls.HtmlInputCheckBox.ServerChange** event. This method allows you to handle the event directly.

This event is raised when the Web page is submitted to the server and the state of the **System.Web.UI.HtmlControls.HtmlInputCheckBox** control changes state from the previous post. This method allows you to override the base

1 implementation and provide a custom handler for the event. A **System.EventArgs**  
2 that contains event information.

### 3 IPostBackDataHandler.LoadPostData

4  
5 [C#] bool IPostBackDataHandler.LoadPostData(string postDataKey,  
6 NameValueCollection postCollection);

7 [C++] bool IPostBackDataHandler::LoadPostData(String\* postDataKey,  
8 NameValueCollection\* postCollection);

9 [VB] Function LoadPostData(ByVal postDataKey As String, ByVal  
10 postCollection As NameValueCollection) As Boolean Implements  
11 IPostBackDataHandler.LoadPostData

12 [JScript] function IPostBackDataHandler.LoadPostData(postDataKey : String,  
13 postCollection : NameValueCollection) : Boolean;

### 14 IPostBackDataHandler.RaisePostDataChangedEvent

15  
16 [C#] void IPostBackDataHandler.RaisePostDataChangedEvent();

17 [C++] void IPostBackDataHandler::RaisePostDataChangedEvent();

18 [VB] Sub RaisePostDataChangedEvent() Implements

19 IPostBackDataHandler.RaisePostDataChangedEvent

20 [JScript] function IPostBackDataHandler.RaisePostDataChangedEvent();

21 HtmlInputControl class (System.Web.UI.HtmlControls)

22 TrackViewState

23  
24  
25 *Description*

Serves as the abstract base class that defines the methods, properties, and events common to all HTML input controls, such as the , , and elements.

The **System.Web.UI.HtmlControls.HtmlInputControl** class cannot be instantiated directly. Instead, this class is inherited by other classes, such as the **System.Web.UI.HtmlControls.HtmlInputText**, **System.Web.UI.HtmlControls.HtmlInputButton**, **System.Web.UI.HtmlControls.HtmlInputRadioButton**, **System.Web.UI.HtmlControls.HtmlInputCheckBox**, **System.Web.UI.HtmlControls.HtmlInputImage**, **System.Web.UI.HtmlControls.HtmlInputHidden**, and **System.Web.UI.HtmlControls.HtmlInputFile** classes, to provide common basic functionality.

**HtmlInputControl**

*Example Syntax:*

**TrackViewState**

[C#]                    public                    **HtmlInputControl**(string                    type);

[C++]                    public:                    **HtmlInputControl**(String\*                    type);

[VB]                    Public                    Sub                    New(ByVal                    type                    As                    String)

[JScript]                    public                    function                    **HtmlInputControl**(type                    :                    String);

### *Description*

Initializes a new instance of the **System.Web.UI.HtmlControls.HtmlInputControl** class.

The following table shows initial property values for an instance of **System.Web.UI.HtmlControls.HtmlInputControl** . The type of input control.

Attributes  
ChildControlsCreated  
ClientID  
Context  
Controls  
Disabled  
EnableViewState  
Events  
HasChildViewState  
ID  
IsTrackingViewState  
Name  
TrackViewState

*Description*

Gets or sets the unique identifier name for the **System.Web.UI.HtmlControls.HtmlInputControl** .

Use the **System.Web.UI.HtmlControls.HtmlInputControl.Name** property to determine the unique identifier name for an **System.Web.UI.HtmlControls.HtmlInputControl** . In this implementation, the **get** accessor returns the value of the **System.Web.UI.Control.UniqueID** property. However, the **set** accessor does not assign a value to this property.



1 NamingContainer  
 2 Page  
 3 Parent  
 4 Site  
 5 Style  
 6 TagName  
 7 TemplateSourceDirectory  
 8 Type  
 9 TrackViewState

10  
 11  
 12 *Description*

13 Gets the type of an **System.Web.UI.HtmlControls.HtmlInputControl** .  
 14 Use this property get the type of an  
 15 **System.Web.UI.HtmlControls.HtmlInputControl** .

16 UniqueID  
 17 Value  
 18 TrackViewState

19  
 20  
 21 *Description*

22 Gets or sets the contents of an  
 23 **System.Web.UI.HtmlControls.HtmlInputControl** .  
 24 Use this property to determine the contents of an  
 25 **System.Web.UI.HtmlControls.HtmlInputControl** . This property can also be

used to programmatically set the contents of an

**System.Web.UI.HtmlControls.HtmlInputControl .**

ViewState

ViewStateIgnoresCase

Visible

RenderAttributes

[C#] protected override void RenderAttributes(HtmlTextWriter writer);

[C++] protected: void RenderAttributes(HtmlTextWriter\* writer);

[VB] Overrides Protected Sub RenderAttributes(ByVal writer As  
HtmlTextWriter)

[JScript] protected override function RenderAttributes(writer : HtmlTextWriter);

#### *Description*

HtmlInputFile class (System.Web.UI.HtmlControls)

TrackViewState

#### *Description*

Allows programmatic access to the HTML element on the server.

Use the **System.Web.UI.HtmlControls.HtmlInputFile** server control to handle uploading binary or text files from a browser client to the server. File upload works with Microsoft Internet Explorer version 3.02 or later.

HtmlInputFile

*Example Syntax:*



1 Use this property to specify the file type that can be uploaded to the server.  
2 For example, to restrict the selection to images, set this property to "image/\*".

3 Attributes

4 ChildControlsCreated

5 ClientID

6 Context

7 Controls

8 Disabled

9 EnableViewState

10 Events

11 HasChildViewState

12 ID

13 IsTrackingViewState

14 MaxLength

15 TrackViewState

16  
17  
18 *Description*

19 Gets or sets the maximum length of the file path for the file to upload from  
20 the client machine.

21 Use this property to specify a limit for the number of characters that can be  
22 entered for the path to the file to upload.

23 Name

24 NamingContainer

25 Page

1 Parent  
2 PostedFile  
3 TrackViewState  
4  
5

6 *Description*

7 Gets access to the uploaded file specified by a client.  
8 Browser security restrictions prevent this value from being maintained  
9 across multiple requests.

10 Site  
11 Size  
12 TrackViewState  
13  
14

15 *Description*

16 Gets or sets the width of the text box in which the file path is entered.  
17 Use this property to specify the width of the text box in which to enter the  
18 file path.

19 Style  
20 TagName  
21 TemplateSourceDirectory  
22 Type  
23 UniqueID  
24 Value  
25 ViewState

1 ViewStateIgnoresCase

2 Visible

3 IPostBackDataHandler.LoadPostData

4

5 [C#] bool IPostBackDataHandler.LoadPostData(string postDataKey,  
6 NameValueCollection postCollection);

7 [C++] bool IPostBackDataHandler::LoadPostData(String\* postDataKey,  
8 NameValueCollection\* postCollection);

9 [VB] Function LoadPostData(ByVal postDataKey As String, ByVal  
10 postCollection As NameValueCollection) As Boolean Implements  
11 IPostBackDataHandler.LoadPostData

12 [JScript] function IPostBackDataHandler.LoadPostData(postDataKey : String,  
13 postCollection : NameValueCollection) : Boolean;

14 IPostBackDataHandler.RaisePostDataChangedEvent

15

16 [C#] void IPostBackDataHandler.RaisePostDataChangedEvent();

17 [C++] void IPostBackDataHandler::RaisePostDataChangedEvent();

18 [VB] Sub RaisePostDataChangedEvent() Implements  
19 IPostBackDataHandler.RaisePostDataChangedEvent

20 [JScript] function IPostBackDataHandler.RaisePostDataChangedEvent();

21 HtmlInputHidden class (System.Web.UI.HtmlControls)

22 TrackViewState

23

24

25 *Description*

1 Allows programmatic access to the HTML element on the server.

2 You can use the **System.Web.UI.HtmlControls.HtmlInputHidden**

3 control to embed non-visible information within a

4 Top of Form

5 element. This information is sent when the Web page is posted back to the

6 server.

7 **HtmlInputHidden**

8 *Example Syntax:*

9 **TrackViewState**

10

11 [C#] public **HtmlInputHidden()**;

12 [C++] public: **HtmlInputHidden()**;

13 [VB] Public Sub **New()**

14 [JScript] public function **HtmlInputHidden()**;

15

16 *Description*

17 Initializes a new instance of the

18 **System.Web.UI.HtmlControls.HtmlInputHidden** class.

19 **Attributes**

20 **ChildControlsCreated**

21 **ClientID**

22 **Context**

23 **Controls**

24 **Disabled**

25 **EnableViewState**

TOP SECRET

- 1 Events
- 2 HasChildViewState
- 3 ID
- 4 IsTrackingViewState
- 5 Name
- 6 NamingContainer
- 7 Page
- 8 Parent
- 9 Site
- 10 Style
- 11 TagName
- 12 TemplateSourceDirectory
- 13 Type
- 14 UniqueID
- 15 Value
- 16 ViewState
- 17 ViewStateIgnoresCase
- 18 Visible
- 19 TrackViewState

22 *Description*

23 Occurs when the

24 **System.Web.UI.HtmlControls.HtmlInputControl.Value** property is changed on

25 the server.



The **System.Web.UI.HtmlControls.HtmlInputHidden.ServerChange** event is raised when the **System.Web.UI.HtmlControls.HtmlInputControl.Value** property is changed on the server.

#### OnPreRender

[C#] protected override void OnPreRender(EventArgs e);  
[C++] protected: void OnPreRender(EventArgs\* e);  
[VB] Overrides Protected Sub OnPreRender(ByVal e As EventArgs)  
[JScript] protected override function OnPreRender(e : EventArgs);

#### Description

#### OnServerChange

[C#] protected virtual void OnServerChange(EventArgs e);  
[C++] protected: virtual void OnServerChange(EventArgs\* e);  
[VB] Overridable Protected Sub OnServerChange(ByVal e As EventArgs)  
[JScript] protected function OnServerChange(e : EventArgs);

#### Description

Raises the **System.Web.UI.HtmlControls.HtmlInputHidden.ServerChange** event.

The **System.Web.UI.HtmlControls.HtmlInputHidden.ServerChange** event is raised when the

1 **System.Web.UI.HtmlControls.HtmlInputControl.Value** property is changed on  
2 the server. A **System.EventArgs** that contains event data.

3 **IPostBackDataHandler.LoadPostData**

4  
5 [C#] bool IPostBackDataHandler.LoadPostData(string postDataKey,  
6 NameValueCollection postCollection);

7 [C++] bool IPostBackDataHandler::LoadPostData(String\* postDataKey,  
8 NameValueCollection\* postCollection);

9 [VB] Function LoadPostData(ByVal postDataKey As String, ByVal  
10 postCollection As NameValueCollection) As Boolean Implements  
11 IPostBackDataHandler.LoadPostData

12 [JScript] function IPostBackDataHandler.LoadPostData(postDataKey : String,  
13 postCollection : NameValueCollection) : Boolean;

14 **IPostBackDataHandler.RaisePostDataChangedEvent**

15  
16 [C#] void IPostBackDataHandler.RaisePostDataChangedEvent();

17 [C++] void IPostBackDataHandler::RaisePostDataChangedEvent();

18 [VB] Sub RaisePostDataChangedEvent() Implements

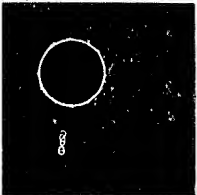
19 IPostBackDataHandler.RaisePostDataChangedEvent

20 [JScript] function IPostBackDataHandler.RaisePostDataChangedEvent();

21 **HtmlInputImage** class (System.Web.UI.HtmlControls)

22 **TrackViewState**

23  
24  
25 *Description*

Allows programmatic access to the HTML  element on the server.

Use the **System.Web.UI.HtmlControls.HtmlInputImage** control to create a button that displays an image. You can programmatically control the action associated with the **System.Web.UI.HtmlControls.HtmlInputImage** control by providing an event handler for the **System.Web.UI.HtmlControls.HtmlInputImage.ServerClick** event.

**HtmlInputImage**

*Example Syntax:*

**TrackViewState**

[C#]	public		<b>HtmlInputImage();</b>
[C++]	public:		<b>HtmlInputImage();</b>
[VB]	Public	Sub	<b>New()</b>
[JScript]	public	function	<b>HtmlInputImage();</b>

*Description*

Initializes a new instance of the **System.Web.UI.HtmlControls.HtmlInputImage** class.

**Align**

**TrackViewState**

[C#]	public	string	<b>Align</b>	{get; set;}
[C++]	public:	__property	String*	get_Align();public: __property void set_Align(String*);

```

1  [VB]          Public          Property      Align      As      String
2  [JScript] public function get Align() : String;public function set Align(String);

```

#### *Description*

Gets or sets the alignment of the **System.Web.UI.HtmlControls.HtmlInputImage** control in relation to other elements on the Web page.

Use the **System.Web.UI.HtmlControls.HtmlInputImage.Align** property to specify the alignment of the **System.Web.UI.HtmlControls.HtmlInputImage** control in relation to the other elements on the Web page. The following table lists the possible values for this property.

Alt

TrackViewState

```

15 [C#]          public          string          Alt          {get;          set;}

```

```

16 [C++] public:  __property  String*  get_Alt();public:  __property  void
17 set_Alt(String*);

```

```

18 [VB]          Public          Property      Alt          As      String

```

```

19 [JScript] public function get Alt() : String;public function set Alt(String);

```

#### *Description*

Gets or sets the alternative text that the browser displays if the image is unavailable or has not been downloaded.

Use the **System.Web.UI.HtmlControls.HtmlInputImage.Alt** property to specify the text to display when the specified image is not available or has not

1 been downloaded. You can also use this property to programmatically determine  
2 the specified alternative text.

3       Attributes

4       Border

5       TrackViewState

6  
7  
8 *Description*

9       Gets       or       sets       the       border       width       for       the  
10 **System.Web.UI.HtmlControls.HtmlInputImage** control.

11       Use       the       **System.Web.UI.HtmlControls.HtmlInputImage.Border**  
12 property       to       specify       the       border       width       for       the  
13 **System.Web.UI.HtmlControls.HtmlInputImage** control.

14       CausesValidation

15       TrackViewState

16  
17 [C#]       public       bool       CausesValidation       {get;       set;}

18 [C++] public: \_\_property bool get\_CausesValidation();public: \_\_property void  
19 set\_CausesValidation(bool);

20 [VB]       Public       Property       CausesValidation       As       Boolean

21 [JScript] public function get CausesValidation() : Boolean;public function set  
22 CausesValidation(Boolean);

23  
24 *Description*  
25

1 Gets or sets a value indicating whether validation is performed when the  
2 **System.Web.UI.HtmlControls.HtmlInputImage** control is clicked.

3 By default, page validation is performed when an  
4 **System.Web.UI.HtmlControls.HtmlInputImage** control is clicked. Page  
5 validation determines whether the input controls associated with a validation  
6 control on the page all pass the validation rules specified by the validation control.

7 ChildControlsCreated

8 ClientID

9 Context

10 Controls

11 Disabled

12 EnableViewState

13 Events

14 HasChildViewState

15 ID

16 IsTrackingViewState

17 Name

18 NamingContainer

19 Page

20 Parent

21 Site

22 Src

23 TrackViewState

24

25

1  
2  
3 *Description*

4 Gets or sets the location of the image file.

5 Use the **System.Web.UI.HtmlControls.HtmlInputImage.Src** property to  
6 specify the location of the image to display in the  
7 **System.Web.UI.HtmlControls.HtmlInputImage** control. If the image is not  
8 available, the text specified in the  
9 **System.Web.UI.HtmlControls.HtmlInputImage.Alt** property is displayed.

10 Style

11 TagName

12 TemplateSourceDirectory

13 Type

14 UniqueID

15 Value

16 ViewState

17 ViewStateIgnoresCase

18 Visible

19 TrackViewState

20  
21  
22 *Description*

23 Occurs on the server when the user clicks an  
24 **System.Web.UI.HtmlControls.HtmlInputImage** control.

The **System.Web.UI.HtmlControls.HtmlInputImage.ServerClick** event is raised when the user clicks an **System.Web.UI.HtmlControls.HtmlInputImage** control.

#### OnPreRender

[C#] protected override void OnPreRender(EventArgs e);  
[C++] protected: void OnPreRender(EventArgs\* e);  
[VB] Overrides Protected Sub OnPreRender(ByVal e As EventArgs)  
[JScript] protected override function OnPreRender(e : EventArgs);

#### Description

##### OnServerClick

[C#] protected virtual void OnServerClick(ImageClickEventArgs e);  
[C++] protected: virtual void OnServerClick(ImageClickEventArgs\* e);  
[VB] Overridable Protected Sub OnServerClick(ByVal e As ImageClickEventArgs)  
[JScript] protected function OnServerClick(e : ImageClickEventArgs);

#### Description

Raises the **System.Web.UI.HtmlControls.HtmlInputImage.ServerClick** event.

The **System.Web.UI.HtmlControls.HtmlInputImage.ServerClick** event is raised when the user clicks an



1 **System.Web.UI.HtmlControls.HtmlInputImage** control. A

2 **System.Web.UI.ImageClickEventArgs** that contains event data.

3 **RenderAttributes**

5 [C#] protected override void RenderAttributes(HtmlTextWriter writer);

6 [C++] protected: void RenderAttributes(HtmlTextWriter\* writer);

7 [VB] Overrides Protected Sub RenderAttributes(ByVal writer As  
8 HtmlTextWriter)

9 [JScript] protected override function RenderAttributes(writer : HtmlTextWriter);

11 *Description*

12 **IPostBackDataHandler.LoadPostData**

14 [C#] bool IPostBackDataHandler.LoadPostData(string postDataKey,  
15 NameValueCollection postCollection);

16 [C++] bool IPostBackDataHandler::LoadPostData(String\* postDataKey,  
17 NameValueCollection\* postCollection);

18 [VB] Function LoadPostData(ByVal postDataKey As String, ByVal  
19 postCollection As NameValueCollection) As Boolean Implements  
20 IPostBackDataHandler.LoadPostData

21 [JScript] function IPostBackDataHandler.LoadPostData(postDataKey : String,  
22 postCollection : NameValueCollection) : Boolean;

23 **IPostBackDataHandler.RaisePostDataChangedEvent**

25 [C#] void IPostBackDataHandler.RaisePostDataChangedEvent();

1 [C++] void IPostBackDataHandler::RaisePostDataChangedEvent();

2 [VB] Sub RaisePostDataChangedEvent() Implements

3 IPostBackDataHandler.RaisePostDataChangedEvent

4 [JScript] function IPostBackDataHandler.RaisePostDataChangedEvent();

5 IPostBackEventHandler.RaisePostBackEvent

7 [C#] void IPostBackEventHandler.RaisePostBackEvent(string eventArgument);

8 [C++] void IPostBackEventHandler::RaisePostBackEvent(String\*  
9 eventArgument);

10 [VB] Sub RaisePostBackEvent(ByVal eventArgument As String) Implements

11 IPostBackEventHandler.RaisePostBackEvent

12 [JScript] function IPostBackEventHandler.RaisePostBackEvent(eventArgument :  
13 String);

14 HtmlInputRadioButton class (System.Web.UI.HtmlControls)

15 TrackViewState

## 18 *Description*

19 Allows programmatic access to the HTML ☐ element on the server.

20 Use the **System.Web.UI.HtmlControls.HtmlInputRadioButton** control  
21 to create a radio button on a Web page. The  
22 **System.Web.UI.HtmlControls.HtmlInputRadioButton** control does not provide  
23 built-in functionality to display a caption for the radio button. To create a caption,  
24 use literal text in the Web page at the desired location. This allows you to control  
25 where the caption is displayed relative to the radio button. For example, if you

1 want to display the caption on the right side of the radio button, declare an  
2 **System.Web.UI.HtmlControls.HtmlInputRadioButton** control followed by the  
3 caption text.

4 **HtmlInputRadioButton**

5 *Example Syntax:*

6 **TrackViewState**

8 [C#] public **HtmlInputRadioButton()**;

9 [C++] public: **HtmlInputRadioButton()**;

10 [VB] Public Sub **New()**

11 [JScript] public function **HtmlInputRadioButton()**;

13 *Description*

14 Initializes a new instance of the  
15 **System.Web.UI.HtmlControls.HtmlInputRadioButton** class.

16 Use this constructor to create and initialize a new instance of the  
17 **System.Web.UI.HtmlControls.HtmlInputRadioButton** class.

18 **Attributes**

19 **Checked**

20 **TrackViewState**

23 *Description*

24 Gets or sets a value indicating whether the  
25 **System.Web.UI.HtmlControls.HtmlInputRadioButton** control is selected.

1 Use the  
2 **System.Web.UI.HtmlControls.HtmlInputRadioButton.Checked** property to  
3 determine whether the **System.Web.UI.HtmlControls.HtmlInputRadioButton**  
4 control is selected. You can also use this property to programmatically specify  
5 whether the control is selected. If you have a group of  
6 **System.Web.UI.HtmlControls.HtmlInputRadioButton** controls, you need to  
7 iterate through each control and test the  
8 **System.Web.UI.HtmlControls.HtmlInputRadioButton.Checked** property of  
9 each control individually.

10 ChildControlsCreated

11 ClientID

12 Context

13 Controls

14 Disabled

15 EnableViewState

16 Events

17 HasChildViewState

18 ID

19 IsTrackingViewState

20 Name

21 TrackViewState

22  
23  
24 *Description*  
25

1 Gets or sets the name of the group that the current instance of the  
2 **System.Web.UI.HtmlControls.HtmlInputRadioButton** class is associated with.

3 Group multiple **System.Web.UI.HtmlControls.HtmlInputRadioButton**  
4 controls together by specifying a common value for the  
5 **System.Web.UI.HtmlControls.HtmlInputControl.Name** property of each radio  
6 button you want to include in the group. When you group  
7 **System.Web.UI.HtmlControls.HtmlInputRadioButton** controls together, only  
8 one radio button in the group can be selected at a time. The  
9 **System.Web.UI.HtmlControls.HtmlInputRadioButton.Checked** property of  
10 the selected control is set to **true** , while the same property is set to **false** for all  
11 other check boxes in the group.

12 NamingContainer

13 Page

14 Parent

15 Site

16 Style

17 TagName

18 TemplateSourceDirectory

19 Type

20 UniqueID

21 Value

22 TrackViewState

23  
24  
25 *Description*

1 Gets or sets the quantity associated with the  
2 **System.Web.UI.HtmlControls.HtmlInputRadioButton** control.

3 Use the **System.Web.UI.HtmlControls.HtmlInputRadioButton.Value**  
4 property to associate a quantity with the  
5 **System.Web.UI.HtmlControls.HtmlInputRadioButton** control. The quantity is  
6 not restricted to a numeric value and can be any valid string. This is useful when  
7 you have multiple radio buttons and need perform a calculation based on the  
8 selection. For example, you can have radio buttons that represent different  
9 shipping methods. You can store the shipping cost in the  
10 **System.Web.UI.HtmlControls.HtmlInputRadioButton.Value** property. When  
11 the user selects the shipping method, you add the appropriate amount to the sales  
12 total.

13 ViewState

14 ViewStateIgnoresCase

15 Visible

16 TrackViewState

17  
18  
19 *Description*

20 Occurs when the value of the  
21 **System.Web.UI.HtmlControls.HtmlInputRadioButton.Checked** property of  
22 the **System.Web.UI.HtmlControls.HtmlInputRadioButton** control changes  
23 between posts to the server.

24 The  
25 **System.Web.UI.HtmlControls.HtmlInputRadioButton.ServerChange** event is

1 raised when the  
2 **System.Web.UI.HtmlControls.HtmlInputRadioButton.Checked** property of  
3 the **System.Web.UI.HtmlControls.HtmlInputRadioButton** control changes  
4 value between posts to the server. This allows you to create a custom event  
5 handler that performs a specific set of instructions (such as data validation) when  
6 the event is raised.

#### 7 OnPreRender

8  
9 [C#] protected override void OnPreRender(EventArgs e);  
10 [C++] protected: void OnPreRender(EventArgs\* e);  
11 [VB] Overrides Protected Sub OnPreRender(ByVal e As EventArgs)  
12 [JScript] protected override function OnPreRender(e : EventArgs);  
13

#### 14 *Description*

#### 15 OnServerChange

16  
17 [C#] protected virtual void OnServerChange(EventArgs e);  
18 [C++] protected: virtual void OnServerChange(EventArgs\* e);  
19 [VB] Overridable Protected Sub OnServerChange(ByVal e As EventArgs)  
20 [JScript] protected function OnServerChange(e : EventArgs);  
21

#### 22 *Description*

23 Raises the  
24 **System.Web.UI.HtmlControls.HtmlInputRadioButton.ServerChange** event.  
25 This allows you to create a custom event handler when the event is raised.

The **System.Web.UI.HtmlControls.HtmlInputRadioButton.ServerChange** event is raised when the **System.Web.UI.HtmlControls.HtmlInputRadioButton.Checked** property of the **System.Web.UI.HtmlControls.HtmlInputRadioButton** control changes values between posts to the server. This allows you to create a custom event handler that performs a specific set of instructions (such as data validation) when the event is raised. A **System.EventArgs** that contains the event data.

#### RenderAttributes

```
[C#] protected override void RenderAttributes(HtmlTextWriter writer);
[C++] protected: void RenderAttributes(HtmlTextWriter* writer);
[VB] Overrides Protected Sub RenderAttributes(ByVal writer As
HtmlTextWriter)
[JScript] protected override function RenderAttributes(writer : HtmlTextWriter);
```

#### *Description*

#### IPostBackDataHandler.LoadPostData

```
[C#] bool IPostBackDataHandler.LoadPostData(string postDataKey,
NameValueCollection postCollection);
[C++] bool IPostBackDataHandler::LoadPostData(String* postDataKey,
NameValueCollection* postCollection);
[VB] Function LoadPostData(ByVal postDataKey As String, ByVal
```





[illegible]

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25

4	[C#]	public	HtmlInputText();
5	[C++]	public:	HtmlInputText();
6	[VB]	Public	Sub New()
7	[JScript] public function HtmlInputText(); Initializes a new instance of the		
8	<b>System.Web.UI.HtmlControls.HtmlInputText</b>		class.

10	Description
----	-------------

11        Initializes a new instance of the  
12        **System.Web.UI.HtmlControls.HtmlInputText** class using default values.

13 This constructor creates a **text** type text box control.

14 | `HtmlInputText`

15 *Example Syntax:*

16 | TrackViewState

18	[C#]	public	HtmlInputText(string type);			
19	[C++]	public:	HtmlInputText(String* type);			
20	[VB]	Public	Sub	New(ByVal	type	As String)
21	[JScript]	public	function	HtmlInputText(type	:	String);

23 | *Description*

1        Initializes        a        new        instance        of        the  
2        **System.Web.UI.HtmlControls.HtmlInputText** class using the specified input  
3        control type.

4        This constructor allows you to create a specific type of text box control,  
5        such as **password** . Only the **password** or **text** type is currently supported. This  
6        constructor is designed to allow you to create other text box types if they become  
7        available. The type of input control.

8        Attributes

9        ChildControlsCreated

10       ClientID

11       Context

12       Controls

13       Disabled

14       EnableViewState

15       Events

16       HasChildViewState

17       ID

18       IsTrackingViewState

19       MaxLength

20       TrackViewState

21  
22  
23       *Description*

24       Gets or sets the maximum number of characters that can be entered in the  
25       text box.

1        Use    the    **System.Web.UI.HtmlControls.HtmlInputText.MaxLength**  
2 property to specify or determine the maximum number of characters that can be  
3 entered in the text box.

4        Name

5        NamingContainer

6        Page

7        Parent

8        Site

9        Size

10       TrackViewState

11  
12  
13       *Description*

14       Gets or sets the width of the text box.

15       Use the **System.Web.UI.HtmlControls.HtmlInputText.Size** property to  
16 specify or determine the width of the textbox, in characters.

17       Style

18       TagName

19       TemplateSourceDirectory

20       Type

21       UniqueID

22       Value

23       TrackViewState

24

25

### *Description*

Gets or sets the contents of the text box.

Use the **System.Web.UI.HtmlControls.HtmlInputText.Value** property to programmatically determine the text entered by the user into the text box. You can also use this property to provide default text for the text box.

ViewState

ViewStateIgnoresCase

Visible

TrackViewState

### *Description*

Occurs when the **System.Web.UI.HtmlControls.HtmlInputText.Value** property is changed on the server.

The **System.Web.UI.HtmlControls.HtmlInputText.ServerChange** event is raised when the **System.Web.UI.HtmlControls.HtmlInputText.Value** property is changed on the server.

OnPreRender

[C#]      protected      override      void      OnPreRender(EventArgs      e);

[C++]      protected:      void      OnPreRender(EventArgs\*      e);

[VB]      Overrides      Protected      Sub      OnPreRender(ByVal      e      As      EventArgs)

[JScript]      protected      override      function      OnPreRender(e      :      EventArgs);

1  
2 *Description*

3       OnServerChange

4  
5 [C#]     protected     virtual     void     OnServerChange(EventArgs    e);

6 [C++]    protected:    virtual     void     OnServerChange(EventArgs\*   e);

7 [VB]    Overridable Protected Sub OnServerChange(ByVal e As EventArgs)

8 [JScript]   protected     function    OnServerChange(e     :    EventArgs);

9  
10 *Description*

11       Raises the **System.Web.UI.HtmlControls.HtmlInputText.ServerChange**  
12 event.

13       The **System.Web.UI.HtmlControls.HtmlInputText.ServerChange** event  
14 is raised when the **System.Web.UI.HtmlControls.HtmlInputText.Value**  
15 property is changed on the server. A **System.EventArgs** that contains event data.

16       RenderAttributes

17  
18 [C#]    protected    override   void   RenderAttributes(HtmlTextWriter   writer);

19 [C++]    protected:    void       RenderAttributes(HtmlTextWriter\*     writer);

20 [VB]    Overrides   Protected   Sub   RenderAttributes(ByVal    writer   As  
21 HtmlTextWriter)

22 [JScript] protected override function RenderAttributes(writer : HtmlTextWriter);

23  
24 *Description*

```

1      IPostBackDataHandler.LoadPostData
2
3      [C#]      bool      IPostBackDataHandler.LoadPostData(string      postDataKey,
4      NameValueCollection      postCollection);
5      [C++]      bool      IPostBackDataHandler::LoadPostData(String*      postDataKey,
6      NameValueCollection*      postCollection);
7      [VB]      Function      LoadPostData(ByVal      postDataKey      As      String,      ByVal
8      postCollection      As      NameValueCollection)      As      Boolean      Implements
9      IPostBackDataHandler.LoadPostData
10     [JScript]      function      IPostBackDataHandler.LoadPostData(postDataKey : String,
11     postCollection : NameValueCollection) : Boolean;
12
13     IPostBackDataHandler.RaisePostDataChangedEvent
14
15     [C#]      void      IPostBackDataHandler.RaisePostDataChangedEvent();
16     [C++]      void      IPostBackDataHandler::RaisePostDataChangedEvent();
17     [VB]      Sub      RaisePostDataChangedEvent()      Implements
18     IPostBackDataHandler.RaisePostDataChangedEvent
19     [JScript]      function      IPostBackDataHandler.RaisePostDataChangedEvent();
20
21     HtmlSelect class (System.Web.UI.HtmlControls)
22
23     TrackViewState
24
25

```

*Description*

Allows programmatic access to the HTML

## **System.Web.UI.WebControls**

### *Description*

The **System.Web.UI.WebControls** namespace is a collection of classes that allow you to create Web server controls on a Web page. Web controls run on the server and include form controls such as buttons and text boxes, as well as special purpose controls such as a calendar. This allows you to programmatically control these elements on a Web page. Web controls are more abstract than HTML controls. Their object model does not necessarily reflect HTML syntax.

**AdCreatedEventArgs** class (**System.Web.UI.WebControls**)

### *Description*

Provides data for the **System.Web.UI.WebControls.AdRotator.AdCreated** event of the **System.Web.UI.WebControls.AdRotator** control. This class cannot be inherited.

The **System.Web.UI.WebControls.AdRotator.AdCreated** event is raised when the **System.Web.UI.WebControls.AdRotator** displays an advertisement on the page.

Constructors:

**AdCreatedEventArgs**

*Example Syntax:*



```

1
2 [C#] public AdCreatedEventArgs(IDictionary adProperties);
3 [C++] public: AdCreatedEventArgs(IDictionary* adProperties);
4 [VB] Public Sub New(ByVal adProperties As IDictionary)
5 [JScript] public function AdCreatedEventArgs(adProperties : IDictionary);
6

```

### *Description*

Initializes a new instance of the **System.Web.UI.WebControls.AdCreatedEventArgs** class.

Use this constructor to create and initialize a new instance of the **System.Web.UI.WebControls.AdCreatedEventArgs** class. A **System.Collections.IDictionary** containing the advertisement properties from the XML file.

Properties:

AdProperties

```

17 [C#] public IDictionary AdProperties {get;}
18 [C++] public: __property IDictionary* get_AdProperties();
19 [VB] Public ReadOnly Property AdProperties As IDictionary
20 [JScript] public function get AdProperties() : IDictionary;
21

```

### *Description*

Gets a **System.Collections.IDictionary** object that contains all the advertisement properties for the currently displayed advertisement.

1        Use the  
2        **System.Web.UI.WebControls.AdCreatedEventArgs.AdProperties** property to  
3        get a **System.Collections.IDictionary** object that contains the advertisement  
4        properties for the currently displayed advertisement. The keys and values of the  
5        **System.Collections.IDictionary** object are of type **System.String** .

#### 6        AlternateText

7  
8        [C#] public string AlternateText {get; set;}  
9        [C++] public: \_\_property String\* get\_AlternateText();public: \_\_property void  
10       set\_AlternateText(String\*);  
11       [VB] Public Property AlternateText As String  
12       [JScript] public function get AlternateText() : String;public function set  
13       AlternateText(String);

#### 15       *Description*

16       Gets or sets the alternate text displayed in the  
17       **System.Web.UI.WebControls.AdRotator** control when the advertisement image  
18       is unavailable. Browsers that support the ToolTips feature display this text as a  
19       ToolTip for the advertisement.

20       Use the  
21       **System.Web.UI.WebControls.AdCreatedEventArgs.AlternateText** property to  
22       specify the text to display if the image specified in the  
23       **System.Web.UI.WebControls.AdCreatedEventArgs.ImageUrl** property is not  
24       available. In browsers that support the ToolTips feature, this text also displays as a  
25       ToolTip for the advertisement.

## ImageUrl

[C#] public string ImageUrl {get; set;}

[C++] public: \_\_property String\* get\_ImageUrl();public: \_\_property void  
set\_ImageUrl(String\*);

[VB] Public Property ImageUrl As String

[JScript] public function get ImageUrl() : String;public function set  
ImageUrl(String);

### *Description*

Gets or sets the URL of an image to display in the  
**System.Web.UI.WebControls.AdRotator** control.

Use the **System.Web.UI.WebControls.AdCreatedEventArgs.ImageUrl**  
property to specify the URL of an image to display for an advertisement in the  
**System.Web.UI.WebControls.AdRotator** control. You can use a relative or an  
absolute URL. A relative URL relates the location of the image to the location of  
the Web page without specifying a complete path on the server. The path is  
relative to the location of the Web page. This makes it easier to move the entire  
site to another directory on the server without updating the path to the image in  
code. An absolute URL provides the complete path, so moving the site to another  
directory requires updating the code.

## NavigateUrl

[C#] public string NavigateUrl {get; set;}

[C++] public: \_\_property String\* get\_NavigateUrl();public: \_\_property void

1 set\_NavigateUrl(String\*);

2 [VB] Public Property NavigateUrl As String

3 [JScript] public function get NavigateUrl() : String;public function set

4 NavigateUrl(String);

5  
6 *Description*

7 Gets or sets the Web page to display when the  
8 **System.Web.UI.WebControls.AdRotator** control is clicked.

9 Use the  
10 **System.Web.UI.WebControls.AdCreatedEventArgs.NavigateUrl** property to  
11 specify the Web page to display when the  
12 **System.Web.UI.WebControls.AdRotator** control is clicked.

13 *Methods:*

14 AdCreatedEventHandler delegate (System.Web.UI.WebControls)

15 ToString

16  
17  
18 *Description*

19 Represents the method that will handle the  
20 **System.Web.UI.WebControls.AdRotator.AdCreated** event of an  
21 **System.Web.UI.WebControls.AdRotator** control. The source of the event. An  
22 **System.Web.UI.WebControls.AdCreatedEventArgs** that contains the event  
23 data.  
24  
25

1 The **System.Web.UI.WebControls.AdRotator.AdCreated** event is raised  
2 when the **System.Web.UI.WebControls.AdRotator** displays an advertisement on  
3 the page.

4 AdRotator class (System.Web.UI.WebControls)

5 ToString

6  
7  
8 *Description*

9 Displays an advertisement banner on a Web page.

10 Use the **System.Web.UI.WebControls.AdRotator** control to display a  
11 randomly selected advertisement banner on the Web page. The displayed  
12 advertisement can change whenever the page refreshes.

13 AdRotator

14 *Example Syntax:*

15 ToString

16  
17 [C#] public AdRotator();

18 [C++] public: AdRotator();

19 [VB] Public Sub New()

20 [JScript] public function AdRotator();

21  
22 *Description*

23 Initializes a new instance of the **System.Web.UI.WebControls.AdRotator**  
24 class.

1 Use this constructor to create and initialize a new instance of the  
2 **System.Web.UI.WebControls.AdRotator** class.

3 AccessKey

4 AdvertisementFile

5 ToString

6  
7  
8 *Description*

9 Gets or sets the path to an XML file that contains advertisement  
10 information.

11 Use the **System.Web.UI.WebControls.AdRotator.AdvertisementFile**  
12 property to specify the location of an XML file that contains advertisement  
13 information. The XML file must reside within the same web site. For deployment  
14 and security purposes, you should place the file in the same web application. It is  
15 only possible to access an XML file that is located in a different application on the  
16 same site if the application has sufficient trust.

17 Attributes

18 BackColor

19 BorderColor

20 BorderStyle

21 BorderWidth

22 ChildControlsCreated

23 ClientID

24 Context

25 Controls

1 ControlStyle  
2 ControlStyleCreated  
3 CssClass  
4 Enabled  
5 EnableViewState  
6 Events  
7 Font  
8 ToString

11 *Description*

12 Font property. Has no effect on this control, so hide it.

13 ForeColor

14 HasChildViewState

15 Height

16 ID

17 IsTrackingViewState

18 KeywordFilter

19 ToString

22 *Description*

23 Gets or sets a category keyword to filter for specific types of  
24 advertisements in the XML advertisement file.

1 Each advertisement in the XML advertisement file can be assigned a  
2 category keyword. Use the  
3 **System.Web.UI.WebControls.AdRotator.KeywordFilter** property to filter the  
4 advertisements for the specified keyword. Only advertisements containing the  
5 keyword will be selected for the **System.Web.UI.WebControls.AdRotator**  
6 control. The **System.Web.UI.WebControls.AdRotator.KeywordFilter** property  
7 can be programmatically set to match the profile of the user.

8 NamingContainer

9 Page

10 Parent

11 Site

12 Style

13 TabIndex

14 TagKey

15 TagName

16 Target

17 ToString

18  
19  
20 *Description*

21 Gets or sets the name of the browser window or frame that displays the  
22 contents of the Web page linked to when the  
23 **System.Web.UI.WebControls.AdRotator** control is clicked.

24 Use the **System.Web.UI.WebControls.AdRotator.Target** property to  
25 specify the target window or frame that displays the contents of the Web page



1 linked to when the **System.Web.UI.WebControls.AdRotator** control is clicked.

2 Values must begin with a letter in the range of a to z (case insensitive), except for  
3 the following special values, which begin with an underscore: **\_blank** Renders the  
4 content in a new, unframed window.

5       TemplateSourceDirectory

6       ToolTip

7       UniqueID

8       ViewState

9       ViewStateIgnoresCase

10      Visible

11      Width

12      ToString

13  
14  
15 *Description*

16       Occurs once per round trip to the server after the creation of the control, but  
17 before the page is rendered.

18       This event is raised once per round trip to the server after the creation of the  
19 control, but before the page is rendered.

20       CreateControlCollection

21  
22 [C#] protected override ControlCollection CreateControlCollection();

23 [C++] protected: ControlCollection\* CreateControlCollection();

24 [VB] Overrides Protected Function CreateControlCollection() As

25 ControlCollection

1 [JScript] protected override function CreateControlCollection() :

2 ControlCollection;

3  
4 *Description*

5  
6 OnAdCreated

7  
8 [C#] protected virtual void OnAdCreated(AdCreatedEventArgs e);

9 [C++] protected: virtual void OnAdCreated(AdCreatedEventArgs\* e);

10 [VB] Overridable Protected Sub OnAdCreated(ByVal e As AdCreatedEventArgs)

11 [JScript] protected function OnAdCreated(e : AdCreatedEventArgs);

12  
13 *Description*

14 Raises the **System.Web.UI.WebControls.AdRotator.AdCreated** event  
15 for the **System.Web.UI.WebControls.AdRotator** control.

16 The **System.Web.UI.WebControls.AdRotator.AdCreated** event is raised  
17 on the server after the **System.Web.UI.WebControls.AdRotator** control is  
18 created, but before the page is rendered. If the

19 **System.Web.UI.WebControls.AdRotator.AdvertisementFile** property is set,  
20 the **System.Web.UI.WebControls.AdRotator.AdCreated** event is raised after an  
21 advertisement has been selected by the

22 **System.Web.UI.WebControls.AdRotator** control. An

23 **System.Web.UI.WebControls.AdCreatedEventArgs** that contains event data.

24 OnPreRender

1  
2 [C#] protected override void OnPreRender(EventArgs e);

3 [C++] protected: void OnPreRender(EventArgs\* e);

4 [VB] Overrides Protected Sub OnPreRender(ByVal e As EventArgs)

5 [JScript] protected override function OnPreRender(e : EventArgs);

6  
7 *Description*

8 Gets the advertisement information for rendering by looking up the file data  
9 and/or calling the user event.

10 Randomly picks an advertisement based on the impresssion (weight) value  
11 of the advertisement.

12 **Render**

13  
14 [C#] protected override void Render(HtmlTextWriter writer);

15 [C++] protected: void Render(HtmlTextWriter\* writer);

16 [VB] Overrides Protected Sub Render(ByVal writer As HtmlTextWriter)

17 [JScript] protected override function Render(writer : HtmlTextWriter);

18  
19 *Description*

20 Displays the **System.Web.UI.WebControls.AdRotator** on the client.

21 Displays an image obtained from the ad chosen in

22 **System.Web.UI.WebControls.AdRotator.OnPreRender(System.EventArgs) .**

23 A **System.Web.UI.HtmlTextWriter** that contains the output stream to render on  
24 the client.

25 **BaseCompareValidator** class (System.Web.UI.WebControls)

TrackViewState

*Description*

Serves as the abstract base class for validation controls that perform comparisons.

The **System.Web.UI.WebControls.BaseCompareValidator** class is inherited by validation controls that compare values, such as the **System.Web.UI.WebControls.CompareValidator** and **System.Web.UI.WebControls.RangeValidator** controls.

BaseCompareValidator

*Example Syntax:*

TrackViewState

[C#] protected BaseCompareValidator();

[C++] protected: BaseCompareValidator();

[VB] Protected Sub New()

[JScript] protected function BaseCompareValidator();

AccessKey

Attributes

BackColor

BorderColor

BorderStyle

BorderWidth

ChildControlsCreated

1 ClientID  
2 Context  
3 Controls  
4 ControlStyle  
5 ControlStyleCreated  
6 ControlToValidate  
7 CssClass  
8 CutoffYear  
9 TrackViewState

10  
11  
12 *Description*

13 Gets the maximum year that can be represented by a two-digit year.

14 Use the

15 **System.Web.UI.WebControls.BaseCompareValidator.CutoffYear** property to  
16 determine the maximum year that can be represented by a two-digit year in a 100  
17 year range. For example, if this property contains the value **2029** , you can  
18 represent the years 1930 to 2029 by using a two-digit year. The two-digit year 30  
19 is interpreted as 1930, while 29 is interpreted as 2029. You cannot change the  
20 value of this property directly, however you can change the maximum year that  
21 can be represented by a two-digit year by setting the  
22 **System.Globalization.Calendar.TwoDigitYearMax** property.

23 Display

24 EnableClientScript

25 Enabled



1  
2  
3 *Description*

4 Gets or sets the data type of the values being compared.

5 Use the **System.Web.UI.WebControls.BaseCompareValidator.Type**  
6 property to specify the data type of the two values being compared. Both values  
7 are converted to the specified data type before any comparison is performed.

8 UniqueID

9 ViewState

10 ViewStateIgnoresCase

11 Visible

12 Width

13 AddAttributesToRender

14  
15 [C#] protected override void AddAttributesToRender(HtmlTextWriter writer);

16 [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);

17 [VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As  
18 HtmlTextWriter)

19 [JScript] protected override function AddAttributesToRender(writer :  
20 HtmlTextWriter);

21  
22 *Description*

23 AddAttributesToRender method AddAttributesToRender method

24 CanConvert

```

1
2 [C#] public static bool CanConvert(string text, ValidationDataType type);
3 [C++] public: static bool CanConvert(String* text, ValidationDataType type);
4 [VB] Public Shared Function CanConvert(ByVal text As String, ByVal type As
5 ValidationDataType) As Boolean
6 [JScript] public static function CanConvert(text : String, type :
7 ValidationDataType) : Boolean;
8

```

### 9 *Description*

10 Determines whether the specified string can be converted to the specified  
11 data type.

12 *Return Value:* **true** if the specified data string is convertible to the specified data  
13 type; otherwise, **false** .

14 Use the  
15 **System.Web.UI.WebControls.BaseCompareValidator.CanConvert(System.St**  
16 **ring,System.Web.UI.WebControls.ValidationDataType)** method to determine  
17 whether the specified string can be converted to the specified data type. This  
18 method is commonly used to test whether a string can be converted to a  
19 compatible data type before performing an operation that depends on that data  
20 type. The string to test whether the specified data type conversion is possible. One  
21 of the **System.Web.UI.WebControls.ValidationDataType** enumeration values.

### 22 *Compare*

```

23
24 [C#] protected static bool Compare(string leftText, string rightText,
25 ValidationCompareOperator op, ValidationDataType type);

```



1 [C++] protected: static bool Compare(String\* leftText, String\* rightText,  
 2 ValidationCompareOperator op, ValidationDataType type);  
 3 [VB] Protected Shared Function Compare(ByVal leftText As String, ByVal  
 4 rightText As String, ByVal op As ValidationCompareOperator, ByVal type As  
 5 ValidationDataType) As Boolean  
 6 [JScript] protected static function Compare(leftText : String, rightText : String, op  
 7 : ValidationCompareOperator, type : ValidationDataType) : Boolean;

### 8 *Description*

9 Compares two strings using the specified operator and validation data type.

10 *Return Value:* **true** if the conversion is successful; otherwise **false** . The string  
 11 value on the left side of the operator. The string value on the right side of the  
 12 operator. A **System.Web.UI.WebControls.ValidationCompareOperator** object  
 13 that represents the comparison operation to perform. A  
 14 **System.Web.UI.WebControls.ValidationDataType** object that represents the

15 data type of the operands.  
 16

### 17 *Convert*

18  
 19 [C#] protected static bool Convert(string text, ValidationDataType type, out object  
 20 value);

21 [C++] protected: static bool Convert(String\* text, ValidationDataType type,  
 22 Object\*\* value);

23 [VB] Protected Shared Function Convert(ByVal text As String, ByVal type As  
 24 ValidationDataType, ByRef value As Object) As Boolean

25 [JScript] protected static function Convert(text : String, type :

1 ValidationDataType, value : Object) : Boolean;

3 *Description*

4 Converts the specified text into an object with specified validation data  
5 type.

6 *Return Value:* **true** if the conversion is successful; otherwise **false** . The source  
7 text to convert from. A **System.Web.UI.WebControls.ValidationDataType**  
8 object that represents the data type to convert the text into. A **System.Object** that  
9 contains the conversion result.

10 DetermineRenderUplevel

12 [C#] protected override bool DetermineRenderUplevel();

13 [C++] protected: bool DetermineRenderUplevel();

14 [VB] Overrides Protected Function DetermineRenderUplevel() As Boolean

15 [JScript] protected override function DetermineRenderUplevel() : Boolean;

17 *Description*

18 Determines whether the validation control can be rendered for an uplevel  
19 browser.

20 *Return Value:* **true** if the validation control can be rendered for an uplevel  
21 browser; otherwise, **false** .

22 GetDateElementOrder

24 [C#] protected static string GetDateElementOrder();

25 [C++] protected: static String\* GetDateElementOrder();

1 [VB] Protected Shared Function GetDateElementOrder() As String

2 [JScript] protected static function GetDateElementOrder() : String;

4 *Description*

5 Return the order of date elements for the current culture Return the order of  
6 date elements for the current culture

7 **GetFullYear**

9 [C#] protected static int GetFullYear(int shortYear);

10 [C++] protected: static int GetFullYear(int shortYear);

11 [VB] Protected Shared Function GetFullYear(ByVal shortYear As Integer) As  
12 Integer

13 [JScript] protected static function GetFullYear(shortYear : int) : int;

15 *Description*

16 Generates the four-digit year representation of the specified two-digit year.

17 *Return Value:* The four-digit year representation of the specified two-digit year.

18 Use the

19 **System.Web.UI.WebControls.BaseCompareValidator.GetFullYear(System.In**

20 **t32)** method to generate the four-digit year representation of the specified two-  
21 digit year. The four-digit year representation depends on the value of the

22 **System.Globalization.BaseCompareValidator.CutoffYear** property, which

23 contains the maximum year that can be represented by a two-digit year in a 100  
24 year range. For example, if the

25 **System.Globalization.BaseCompareValidator.CutoffYear** property contains

the value 2029 , the

**System.Web.UI.WebControls.BaseCompareValidator.GetFullYear(System.Int32)** method returns a year between 1930 to 2029. The two-digit year 30 is interpreted as 1930, while 29 is interpreted as 2029. You can change the maximum year that can be represented by a two-digit year by setting the **System.Globalization.Calendar.TwoDigitYearMax** property. A two-digit year.

BaseDataList class (System.Web.UI.WebControls)

Validate

### *Description*

Serves as the abstract base class for data list controls, such as the **System.Web.UI.WebControls.DataList** and **System.Web.UI.WebControls.DataGrid** controls. This class provides methods and properties that are common to all data list controls.

The **System.Web.UI.WebControls.BaseDataList** class provides the common functionality for all data list controls, such as the **System.Web.UI.WebControls.DataList** and **System.Web.UI.WebControls.DataGrid** controls. An instance of this class is not created directly. Instead, data list controls inherit the methods and properties common to all data list controls from this class.

BaseDataList

### *Example Syntax:*

Validate

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#] public BaseDataList();  
[C++] public: BaseDataList();  
[VB] Public Sub New()  
[JScript] public function BaseDataList();
```

*Description*

Initializes a new instance of the **System.Web.UI.WebControls.BaseDataList** class.

A **System.Web.UI.WebControls.BaseDataList** object is not created directly. This constructor is commonly called in the constructor of a derived class to initialize the properties defined in the **System.Web.UI.WebControls.BaseDataList** class.

- AccessKey
- Attributes
- BackColor
- BorderColor
- BorderStyle
- BorderWidth
- CellPadding
- Validate

*Description*

1 Gets or sets the amount of space between the contents of a cell and the  
2 cell's border.

3 Use the **System.Web.UI.WebControls.BaseDataList.CellPadding**  
4 property to control the spacing between the contents of a cell and the cell's border.  
5 The padding amount specified is added to all four sides of a cell. All cells in the  
6 same column of the list control share the same cell width. Therefore, if the content  
7 of one cell is longer than the content other cells in the same column, the padding  
8 amount is applied to the width of longest cell. All other cells in the column will  
9 also share the same cell width. The same is true for all cells in the same row. The  
10 padding amount is applied to the tallest cell in the row, with all cells in the same  
11 row sharing the same cell height. Individual cell sizes cannot be specified.

12 CellSpacing

13 Validate

14  
15 [C#] public virtual int CellSpacing {get; set;}

16 [C++] public: \_\_property virtual int get\_CellSpacing();public: \_\_property virtual  
17 void set\_CellSpacing(int);

18 [VB] Overridable Public Property CellSpacing As Integer

19 [JScript] public function get CellSpacing() : int;public function set  
20 CellSpacing(int);

21  
22 *Description*

23 Gets or sets the amount of space between cells.  
24  
25

1        Use the **System.Web.UI.WebControls.BaseDataList.CellSpacing**  
2        property to control the spacing between adjacent cells in the list control. This  
3        spacing is applied both vertically and horizontally.

4        ChildControlsCreated

5        ClientID

6        Context

7        Controls

8        Validate

9  
10  
11        *Description*

12  
13        ControlStyle

14        ControlStyleCreated

15        CssClass

16        DataKeyField

17        Validate

18  
19  
20        *Description*

21        Gets or sets the primary key field in the data source referenced by  
22        **System.Web.UI.WebControls.BaseDataList.DataSource** .

23        DataKeys

24        Validate

1  
2 [C#] public DataKeyCollection DataKeys {get;}

3 [C++] public: \_\_property DataKeyCollection\* get\_DataKeys();

4 [VB] Public ReadOnly Property DataKeys As DataKeyCollection

5 [JScript] public function get DataKeys() : DataKeyCollection;

6  
7 *Description*

8 Gets a collection that stores the primary key values of each record  
9 (displayed as a row) in the list control.

10 Use the **System.Web.UI.WebControls.BaseDataList.DataKeyCollection**  
11 to access the primary key values of each record (displayed as a row) in the list  
12 control. This property is commonly used to include the primary key field with the  
13 list control without displaying it in the control. The collection is filled with the  
14 values from the field specified by the  
15 **System.Web.UI.WebControls.BaseDataList.DataKeyField** property.

16 DataKeysArray

17 Validate

18  
19 [C#] protected ArrayList DataKeysArray {get;}

20 [C++] protected: \_\_property ArrayList\* get\_DataKeysArray();

21 [VB] Protected ReadOnly Property DataKeysArray As ArrayList

22 [JScript] protected function get DataKeysArray() : ArrayList;

23  
24 *Description*

25 Gets a list of key fields in the data source.



DataMember

Validate

[C#] public string DataMember {get; set;}

[C++] public: \_\_property String\* get\_DataMember();public: \_\_property void  
set\_DataMember(String\*);

[VB] Public Property DataMember As String

[JScript] public function get DataMember() : String;public function set  
DataMember(String);

### *Description*

Gets or sets the specific data member in a multi member

**System.Web.UI.WebControls.BaseDataList.DataSource** to bind to the list  
control.

Use the **System.Web.UI.WebControls.BaseDataList.DataMember**  
property to specify a member from a multi member

**System.Web.UI.WebControls.BaseDataList.DataSource** to bind to the list  
control. For example, if you had a

**System.Web.UI.WebControls.BaseDataList.DataSource** that contains multiple  
tables, you can use the

**System.Web.UI.WebControls.BaseDataList.DataMember** property to specify  
the table to bind to the list control.

DataSource

Validate

1  
2 [C#] public virtual object DataSource {get; set;}

3 [C++] public: \_\_property virtual Object\* get\_DataSource();public: \_\_property

4 virtual void set\_DataSource(Object\*);

5 [VB] Overridable Public Property DataSource As Object

6 [JScript] public function get DataSource() : Object;public function set

7 DataSource(Object);

8  
9 *Description*

10 Gets or sets the source to a list of values used to populate the items within  
11 the control.

12 Lists derived from ICollection can also be used as a DataSource.

13 Enabled

14 EnableViewState

15 Events

16 Font

17 ForeColor

18 GridLines

19 Validate

20  
21  
22 *Description*

23 Gets or sets a value that specifies the grid line style.

24 HasChildViewState

25 Height

1	HorizontalAlign
2	Validate
3	
4	
5	<i>Description</i>
6	Gets or sets the horizontal alignment of the control within its container.
7	ID
8	IsTrackingViewState
9	NamingContainer
10	Page
11	Parent
12	Site
13	Style
14	TabIndex
15	TagKey
16	TagName
17	TemplateSourceDirectory
18	ToolTip
19	UniqueID
20	ViewState
21	ViewStateIgnoresCase
22	Visible
23	Width
24	Validate
25	

1  
2  
3 *Description*

4 Occurs when an item on the list is selected.

5 AddParsedSubObject

6  
7 [C#] protected override void AddParsedSubObject(object obj);

8 [C++] protected: void AddParsedSubObject(Object\* obj);

9 [VB] Overrides Protected Sub AddParsedSubObject(ByVal obj As Object)

10 [JScript] protected override function AddParsedSubObject(obj : Object);

11  
12 *Description*

13 Overridden. The object to add.

14 CreateChildControls

15  
16 [C#] protected override void CreateChildControls();

17 [C++] protected: void CreateChildControls();

18 [VB] Overrides Protected Sub CreateChildControls()

19 [JScript] protected override function CreateChildControls();

20  
21 *Description*

22 Creates a child control using the view state.

23 CreateControlHierarchy

24  
25 [C#] protected abstract void CreateControlHierarchy(bool useDataSource);

1 [C++] protected: virtual void CreateControlHierarchy(bool useDataSource) = 0;  
 2 [VB] MustOverride Protected Sub CreateControlHierarchy(ByVal useDataSource  
 3 As Boolean)  
 4 [JScript] protected abstract function CreateControlHierarchy(useDataSource :  
 5 Boolean);

#### 6 7 *Description*

8 Overridden. **true** to use the  
 9 **System.Web.UI.WebControls.BaseDataList.DataSource**; otherwise, **false**.

10 DataBind

11  
 12 [C#] public override void DataBind();  
 13 [C++] public: void DataBind();  
 14 [VB] Overrides Public Sub DataBind()  
 15 [JScript] public override function DataBind();

#### 16 17 *Description*

18 Binds the control and all of its child controls to the data source specified by  
 19 the **System.Web.UI.WebControls.BaseDataList.DataSource** property.

20 IsBindableType

21  
 22 [C#] public static bool IsBindableType(Type type);  
 23 [C++] public: static bool IsBindableType(Type\* type);  
 24 [VB] Public Shared Function IsBindableType(ByVal type As Type) As Boolean  
 25 [JScript] public static function IsBindableType(type : Type) : Boolean;

## Description

Determines whether the specified data type is bindable to a list control that derives from the **System.Web.UI.WebControls.BaseDataList** class.

**Return Value:** **true** if the specified data type is bindable to a list control that derives from the **System.Web.UI.WebControls.BaseDataList** class; otherwise, **false**.

Use the

**System.Web.UI.WebControls.BaseDataList.IsBindableType(System.Type)**

static method to determine whether the specified data type is bindable to a list control that inherits from the **System.Web.UI.WebControls.BaseDataList** class.

The supported data types are **System.Boolean**, **System.Byte**, **System.SByte**, **System.Int16**, **System.UInt16**, **System.Int32**, **System.UInt32**, **System.Int64**, **System.UInt64**, **System.Char**, **System.Double**, **System.Single**, **System.DateTime**, **System.Decimal**, and string. A **System.Type** object that contains the data type to test.

**OnDataBinding**

[C#] protected override void OnDataBinding(EventArgs e);

[C++] protected: void OnDataBinding(EventArgs\* e);

[VB] Overrides Protected Sub OnDataBinding(ByVal e As EventArgs)

[JScript] protected override function OnDataBinding(e : EventArgs);

## Description

1       Raises the **DataBinding** event of a  
2 **System.Web.UI.WebControls.BaseDataList** .

3       The event is raised when data is bound to the control. An  
4 **System.EventArgs** that contains the event data.

5       OnSelectedIndexChanged

6  
7 [C#] protected virtual void OnSelectedIndexChanged(EventArgs e);

8 [C++] protected: virtual void OnSelectedIndexChanged(EventArgs\* e);

9 [VB] Overridable Protected Sub OnSelectedIndexChanged(ByVal e As  
10 EventArgs)

11 [JScript] protected function OnSelectedIndexChanged(e : EventArgs);

12  
13 *Description*

14       Raises the  
15 **System.Web.UI.WebControls.BaseDataList.SelectedIndexChanged** event of a  
16 **System.Web.UI.WebControls.BaseDataList** . An **System.EventArgs** that  
17 contains the event data.

18       PrepareControlHierarchy

19  
20 [C#] protected abstract void PrepareControlHierarchy();

21 [C++] protected: virtual void PrepareControlHierarchy() = 0;

22 [VB] MustOverride Protected Sub PrepareControlHierarchy()

23 [JScript] protected abstract function PrepareControlHierarchy();

24  
25 *Description*

1 Overridden.  
 2 Render  
 3  
 4 [C#] protected override void Render(HtmlTextWriter writer);  
 5 [C++] protected: void Render(HtmlTextWriter\* writer);  
 6 [VB] Overrides Protected Sub Render(ByVal writer As HtmlTextWriter)  
 7 [JScript] protected override function Render(writer : HtmlTextWriter);  
 8

9 *Description*

10 Displays the control on the client. An **System.Web.UI.HtmlTextWriter**  
 11 that contains the output stream to render on the client.

12 BaseValidator class (System.Web.UI.WebControls)

13 TrackViewState  
 14  
 15

16 *Description*

17 Serves as the abstract base class for validation controls.

18 The **System.Web.UI.WebControls.BaseValidator** class provides the core  
 19 implementation for all validation controls.

20 BaseValidator

21 *Example Syntax:*

22 TrackViewState  
 23

24 [C#] protected BaseValidator();

25 [C++] protected: BaseValidator();



1 [VB] Protected Sub New()

2 [JScript] protected function BaseValidator();

3  
4 *Description*

5       Initializes a new instance of the  
6 **System.Web.UI.WebControls.BaseValidator** class.

7       This constructor is not called directly. Validation controls that inherit from  
8 this class can call this constructor from their own constructors to initialize the base  
9 properties.

10       AccessKey

11       Attributes

12       BackColor

13       BorderColor

14       BorderStyle

15       BorderWidth

16       ChildControlsCreated

17       ClientID

18       Context

19       Controls

20       ControlStyle

21       ControlStyleCreated

22       ControlToValidate

23       TrackViewState

1  
2  
3 *Description*

4 Gets or sets the input control to validate.

5 Use the

6 **System.Web.UI.WebControls.BaseValidator.ControlToValidate** property to  
7 specify the input control to validate. This property must be set to the ID of an  
8 input control for all validation controls except for the  
9 **System.Web.UI.WebControls.CustomValidator** control, which can be left  
10 blank. If you do not specify a valid input control, an exception will be thrown  
11 when the page is rendered. The ID must refer to a control within the same  
12 container as the validation control. It must be in the same page or user control, or  
13 it must be in the same template of a templated control.

14 **CssClass**

15 **Display**

16 **TrackViewState**  
17  
18

19 *Description*

20 Gets or sets the display behavior of the error message in a validation  
21 control.

22 Use the **System.Web.UI.WebControls.BaseValidator.Display** property to  
23 specify the display behavior of the error message in the validation control. The  
24 following table lists the different values that can be used.

25 **EnableClientScript**

## TrackViewState

[C#] public bool EnableClientScript {get; set;}

[C++] public: \_\_property bool get\_EnableClientScript();public: \_\_property void set\_EnableClientScript(bool);

[VB] Public Property EnableClientScript As Boolean

[JScript] public function get EnableClientScript() : Boolean;public function set EnableClientScript(Boolean);

### *Description*

Gets or sets a value indicating whether client-side validation is enabled.

Use the

**System.Web.UI.WebControls.BaseValidator.EnableClientScript** property to specify whether client-side validation is enabled.

Enabled

TrackViewState

[C#] public override bool Enabled {get; set;}

[C++] public: \_\_property virtual bool get\_Enabled();public: \_\_property virtual void set\_Enabled(bool);

[VB] Overrides Public Property Enabled As Boolean

[JScript] public function get Enabled() : Boolean;public function set Enabled(Boolean);

### *Description*

1 Gets or sets a value that indicates whether the validation control is enabled.  
2 Use the **System.Web.UI.WebControls.BaseValidator.Enabled** property  
3 to specify whether the validation control is enabled. You can programmatically  
4 disable the validation control by setting this property to **false** .

5 EnableViewState

6 ErrorMessage

7 TrackViewState

8  
9  
10 *Description*

11 Gets or sets the text for the error message.

12 Use the **System.Web.UI.WebControls.BaseValidator.ErrorMessage**  
13 property to specify the text to display in the validation control when validation  
14 fails. This text is also included in the  
15 **System.Web.UI.WebControls.ValidationSummary** control, if one is placed on  
16 the Web page.

17 Events

18 Font

19 ForeColor

20 TrackViewState

21  
22  
23 *Description*

24 Gets or sets the color of the message displayed when validation fails.  
25

1        Use the **System.Web.UI.WebControls.BaseValidator.ForeColor**  
2 property to specify a custom text color for the message displayed when validation  
3 fails.

4        HasChildViewState

5        Height

6        ID

7        IsTrackingViewState

8        IsValid

9        TrackViewState

10  
11  
12        *Description*

13           Gets or sets a value that indicates whether the associated input control  
14 passes validation.

15           Use the **System.Web.UI.WebControls.BaseValidator.IsValid** property to  
16 determine whether the associated input control passes validation.

17        NamingContainer

18        Page

19        Parent

20        PropertiesValid

21        TrackViewState

22  
23  
24        *Description*  
25

1 Gets a value that indicates whether the control specified by the  
2 **System.Web.UI.WebControls.BaseValidator.ControlToValidate** property is a  
3 valid control.

4 Use the **System.Web.UI.WebControls.BaseValidator.PropertiesValid**  
5 property to determine whether the  
6 **System.Web.UI.WebControls.BaseValidator.ControlToValidate** property  
7 contains a valid input control. To be a valid input control, the  
8 **System.Web.UI.WebControls.BaseValidator.ControlToValidate** property must  
9 set to a control on the page.

10 RenderUplevel

11 TrackViewState

12  
13 [C#] protected bool RenderUplevel {get;}

14 [C++] protected: \_\_property bool get\_RenderUplevel();

15 [VB] Protected ReadOnly Property RenderUplevel As Boolean

16 [JScript] protected function get RenderUplevel() : Boolean;

17  
18 *Description*

19 Gets a value that indicates whether the client's browser supports uplevel  
20 rendering.

21 Use the **System.Web.UI.WebControls.BaseValidator.RenderUplevel**  
22 property to determine whether the client's browser supports uplevel rendering. For  
23 a browser to support uplevel rendering, it must support Microsoft XML Document  
24 Object Model (DOM) version 4 or higher and any version of ECMAScript.

25 Site

1 Style  
2 TabIndex  
3 TagKey  
4 TagName  
5 TemplateSourceDirectory  
6 Text  
7 ToolTip  
8 UniqueID  
9 ViewState  
10 ViewStateIgnoresCase  
11 Visible  
12 Width  
13 AddAttributesToRender

14  
15 [C#] protected override void AddAttributesToRender(HtmlTextWriter writer);

16 [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);

17 [VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As  
18 HtmlTextWriter)

19 [JScript] protected override function AddAttributesToRender(writer :  
20 HtmlTextWriter);

21  
22 *Description*

23 Adds the attributes of this control to the output stream for rendering on the  
24 client. An **System.Web.UI.HtmlTextWriter** that contains the output stream to  
25 render on the client.

## CheckControlValidationProperty

```
[C#] protected void CheckControlValidationProperty(string name, string
propertyName);
[C++] protected: void CheckControlValidationProperty(String* name, String*
propertyName);
[VB] Protected Sub CheckControlValidationProperty(ByVal name As String,
ByVal propertyName As String)
[JScript] protected function CheckControlValidationProperty(name : String,
propertyName : String);
```

### *Description*

Helper function that verifies whether the specified control is on the page and contains validation properties.

The **System.Web.UI.WebControls.BaseValidator.CheckControlValidationProperty(System.String, System.String)** method is a helper function primarily used by the **System.Web.UI.WebControls.BaseValidator.ControlPropertiesValid** method to verify whether the specified control is on the page and contains validation properties. This method does not return a value to report the result. Instead, it throws an exception when verification fails. The control to verify. The property that contains the control name.

### ControlPropertiesValid

```
[C#] protected virtual bool ControlPropertiesValid();
```



1 [C++] protected: virtual bool ControlPropertiesValid();

2 [VB] Overridable Protected Function ControlPropertiesValid() As Boolean

3 [JScript] protected function ControlPropertiesValid() : Boolean;

4  
5 *Description*

6 Helper function that determines whether the control specified by the  
7 **System.Web.UI.WebControls.BaseValidator.ControlToValidate** property is a  
8 valid control.

9 *Return Value:* **true** if the control specified by the  
10 **System.Web.UI.WebControls.BaseValidator.ControlToValidate** property is a  
11 valid control; otherwise, **false** .

12 The  
13 **System.Web.UI.WebControls.BaseValidator.ControlPropertiesValid** method  
14 is a helper function primarily used by the  
15 **System.Web.UI.WebControls.BaseValidator.PropertiesValid** property to  
16 determine whether the  
17 **System.Web.UI.WebControls.BaseValidator.ControlToValidate** property  
18 contains a valid input control. To be a valid input control, the  
19 **System.Web.UI.WebControls.BaseValidator.ControlToValidate** property must  
20 be set to some value and that value must be a control on the page.

21 **DetermineRenderUplevel**

22  
23 [C#] protected virtual bool DetermineRenderUplevel();

24 [C++] protected: virtual bool DetermineRenderUplevel();

25 [VB] Overridable Protected Function DetermineRenderUplevel() As Boolean

1 [JScript] protected function DetermineRenderUplevel() : Boolean;

3 *Description*

4       Helper function that determines whether the validator control can be  
5 rendered for an uplevel browser.

6 *Return Value:* **true** if the validator control can be rendered for an uplevel browser;  
7 otherwise, **false** .

8       The  
9 **System.Web.UI.WebControls.BaseValidator.DetermineRenderUplevel**  
10 method is a helper function used primarily by the  
11 **System.Web.UI.WebControls.BaseValidator.RenderUplevel** property to  
12 determine whether the client's browser supports uplevel rendering. For a browser  
13 to support uplevel rendering, it must support Microsoft XML Document Object  
14 Model (DOM) version 4 or higher and any version of ECMAScript.

15 **EvaluateIsValid**

17 [C#] protected abstract bool EvaluateIsValid();

18 [C++] protected: virtual bool EvaluateIsValid() = 0;

19 [VB] MustOverride Protected Function EvaluateIsValid() As Boolean

20 [JScript] protected abstract function EvaluateIsValid() : Boolean;

22 *Description*

23       When overridden in a derived class, this method contains the code to  
24 determine whether the value in the input control is valid.

25 *Return Value:* **true** if the value in the input control is valid; otherwise, **false** .

1 Use the **System.Web.UI.WebControls.BaseValidator.EvaluateIsValid**  
2 method to determine whether the value in the input control specified by the  
3 **System.Web.UI.WebControls.BaseValidator.ControlToValidate** property is  
4 valid.

#### 5 GetControlRenderID

6  
7 [C#] protected string GetControlRenderID(string name);  
8 [C++] protected: String\* GetControlRenderID(String\* name);  
9 [VB] Protected Function GetControlRenderID(ByVal name As String) As String  
10 [JScript] protected function GetControlRenderID(name : String) : String;

#### 11 *Description*

12 Gets the client ID of the specified control.

13 *Return Value:* The client ID of the specified control.

14 Use the

15  
16 **System.Web.UI.WebControls.BaseValidator.GetControlRenderID(System.St**  
17 **ring)** method to get the client ID of the specified control. The name of the control.  
18 to get the client ID.

#### 19 GetControlValidationValue

20  
21 [C#] protected string GetControlValidationValue(string name);  
22 [C++] protected: String\* GetControlValidationValue(String\* name);  
23 [VB] Protected Function GetControlValidationValue(ByVal name As String) As  
24 String  
25 [JScript] protected function GetControlValidationValue(name : String) : String;

1  
2 *Description*

3 Gets the value associated with the specified input control.

4 *Return Value:* The value associated with the specified input control.

5 Use the

6 **System.Web.UI.WebControls.BaseValidator.GetControlValidationValue(System.String)** method to get the value associated with the specified input control,  
7  
8 regardless of the control type. For example, you can use this method to get the  
9 value in a **System.Web.UI.WebControls.TextBox** control, as well as the value of  
10 the selected item from a **System.Web.UI.WebControls.ListBox** control. The  
11 name of the input control to get the value from.

12 **GetValidationProperty**

13  
14 [C#] public static PropertyDescriptor GetValidationProperty(object component);

15 [C++] public: static PropertyDescriptor\* GetValidationProperty(Object\*  
16 component);

17 [VB] Public Shared Function GetValidationProperty(ByVal component As  
18 Object) As PropertyDescriptor

19 [JScript] public static function GetValidationProperty(component : Object) :  
20 PropertyDescriptor;

21  
22 *Description*

23 Helper function to get the validation property of a control (if it exists).

24 *Return Value:* A **System.ComponentModel.PropertyDescriptor** that contains  
25 the validation property of the control.

1 The  
2 **System.Web.UI.WebControls.BaseValidator.GetValidationProperty(System.**  
3 **Object)** method is a helper function that gets the validation property of the  
4 specified input control. An **System.Object** that represents the control to get the  
5 validation property of.

#### 6 OnInit

7  
8 [C#] protected override void OnInit(EventArgs e);  
9 [C++] protected: void OnInit(EventArgs\* e);  
10 [VB] Overrides Protected Sub OnInit(ByVal e As EventArgs)  
11 [JScript] protected override function OnInit(e : EventArgs);  
12

#### 13 *Description*

14 Registers the validator on the page. An **System.EventArgs** that contains  
15 the event data.

#### 16 OnPreRender

17  
18 [C#] protected override void OnPreRender(EventArgs e);  
19 [C++] protected: void OnPreRender(EventArgs\* e);  
20 [VB] Overrides Protected Sub OnPreRender(ByVal e As EventArgs)  
21 [JScript] protected override function OnPreRender(e : EventArgs);  
22

#### 23 *Description*

24 Checks the client browser and configures the validator for compatibility  
25 prior to rendering.

## OnUnload

[C#] protected override void OnUnload(EventArgs e);  
[C++] protected: void OnUnload(EventArgs\* e);  
[VB] Overrides Protected Sub OnUnload(ByVal e As EventArgs)  
[JScript] protected override function OnUnload(e : EventArgs);

### *Description*

Un-registers the validator on the page.

## RegisterValidatorCommonScript

[C#] protected void RegisterValidatorCommonScript();  
[C++] protected: void RegisterValidatorCommonScript();  
[VB] Protected Sub RegisterValidatorCommonScript()  
[JScript] protected function RegisterValidatorCommonScript();

### *Description*

Registers code on the page for client-side validation.

Use the

**System.Web.UI.WebControls.BaseValidator.RegisterValidatorCommonScript**  
method to register code on the page to perform client-side validation.

## RegisterValidatorDeclaration

[C#] protected virtual void RegisterValidatorDeclaration();  
[C++] protected: virtual void RegisterValidatorDeclaration();

1 [VB] Overridable Protected Sub RegisterValidatorDeclaration()

2 [JScript] protected function RegisterValidatorDeclaration(); Registers an  
3 ECMA Script array declaration used on the page by client-side code.

4  
5 *Description*

6 Registers an ECMA Script array declaration using the array name,

7 **Page\_Validators .**

8 Use the

9 **System.Web.UI.WebControls.BaseValidator.RegisterValidatorDeclaration**

10 method to register an ECMA Script array declaration using the array name,

11 **Page\_Validators .**

12 Render

13  
14 [C#] protected override void Render(HtmlTextWriter writer);

15 [C++] protected: void Render(HtmlTextWriter\* writer);

16 [VB] Overrides Protected Sub Render(ByVal writer As HtmlTextWriter)

17 [JScript] protected override function Render(writer : HtmlTextWriter);

18  
19 *Description*

20 Displays the control on the client. An **System.Web.UI.HtmlTextWriter**  
21 that contains the output stream for rendering on the client.

22 Validate

23  
24 [C#] public void Validate();

25 [C++] public: \_\_sealed void Validate();

1 [VB] NotOverridable Public Sub Validate()

2 [JScript] public function Validate();

3  
4 *Description*

5 Performs validation on the associated input control and updates the  
6 **System.Web.UI.WebControls.BaseValidator.IsValid** property.

7 Use the **System.Web.UI.WebControls.BaseValidator.Validate** method to  
8 perform validation on the associated input control. This method allows you to  
9 programmatically perform validation on the input control. The  
10 **System.Web.UI.WebControls.BaseValidator.IsValid** property is automatically  
11 updated with the validation results.

12 BorderStyle enumeration (System.Web.UI.WebControls)

13 Validate

14  
15  
16 *Description*

17 Specifies the border style of a control.

18 The **System.Web.UI.WebControls.BorderStyle** enumeration represents  
19 the different border style options for a control.

20 Validate

21  
22 [C#] public const BorderStyle Dashed;

23 [C++] public: const BorderStyle Dashed;

24 [VB] Public Const Dashed As BorderStyle

25 [JScript] public var Dashed : BorderStyle;



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

*Description*

A dashed line border.

Validate

```
[C#] public const BorderStyle Dotted;  
[C++] public: const BorderStyle Dotted;  
[VB] Public Const Dotted As BorderStyle  
[JScript] public var Dotted : BorderStyle;
```

*Description*

A dotted line border.

Validate

```
[C#] public const BorderStyle Double;  
[C++] public: const BorderStyle Double;  
[VB] Public Const Double As BorderStyle  
[JScript] public var Double : BorderStyle;
```

*Description*

A double solid line border.

Validate

```
[C#] public const BorderStyle Groove;  
[C++] public: const BorderStyle Groove;
```

1 [VB] Public Const Groove As BorderStyle

2 [JScript] public var Groove : BorderStyle;

3  
4 *Description*

5 A grooved border for a sunken border appearance.

6 Validate

7  
8 [C#] public const BorderStyle Inset;

9 [C++] public: const BorderStyle Inset;

10 [VB] Public Const Inset As BorderStyle

11 [JScript] public var Inset : BorderStyle;

12  
13 *Description*

14 An inset border for a sunken control appearance.

15 Validate

16  
17 [C#] public const BorderStyle None;

18 [C++] public: const BorderStyle None;

19 [VB] Public Const None As BorderStyle

20 [JScript] public var None : BorderStyle;

21  
22 *Description*

23 No border.

24 Validate

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#] public const BorderStyle NotSet;  
[C++] public: const BorderStyle NotSet;  
[VB] Public Const NotSet As BorderStyle  
[JScript] public var NotSet : BorderStyle;
```

*Description*

No set border style.

Validate

```
[C#] public const BorderStyle Outset;  
[C++] public: const BorderStyle Outset;  
[VB] Public Const Outset As BorderStyle  
[JScript] public var Outset : BorderStyle;
```

*Description*

An outset border for a raised control appearance.

Validate

```
[C#] public const BorderStyle Ridge;  
[C++] public: const BorderStyle Ridge;  
[VB] Public Const Ridge As BorderStyle  
[JScript] public var Ridge : BorderStyle;
```

*Description*

MSDN: 2004-09-20

1 A ridged border for a raised border appearance.

2 Validate

3

4 [C#] public const BorderStyle Solid;

5 [C++] public: const BorderStyle Solid;

6 [VB] Public Const Solid As BorderStyle

7 [JScript] public var Solid : BorderStyle;

8

9 *Description*

10 A solid line border.

11 BoundColumn class (System.Web.UI.WebControls)

12 ToString

13

14

15 *Description*

16 A column type for the **System.Web.UI.WebControls.DataGrid** control  
17 that is bound to a field in a data source.

18 Use the **System.Web.UI.WebControls.BoundColumn** column type in a  
19 **System.Web.UI.WebControls.DataGrid** control to display the contents of a field  
20 in the data source in a single column. The field is linked to the  
21 **System.Web.UI.WebControls.BoundColumn** , so any updates in the data source  
22 are reflected in the corresponding cells of the  
23 **System.Web.UI.WebControls.DataGrid** control.

24 ToString

25

```

1
2 [C#] public static readonly string thisExpr;
3 [C++] public: static String* thisExpr;
4 [VB] Public Shared ReadOnly thisExpr As String
5 [JScript] public static var thisExpr : String;
6

```

### *Description*

Represents the string "!". This field is read-only.

Use the **System.Web.UI.WebControls.BoundColumn.thisExpr** field to represent the "!" string.

**BoundColumn**

### *Example Syntax:*

**ToString**

```

15 [C#] public BoundColumn();
16 [C++] public: BoundColumn();
17 [VB] Public Sub New()
18 [JScript] public function BoundColumn();
19

```

### *Description*

Initializes a new instance of the

**System.Web.UI.WebControls.BoundColumn** class.

Use this constructor to create and initialize a new instance of the

**System.Web.UI.WebControls.BoundColumn** class.

**DataField**

ToString

[C#] public virtual string DataField {get; set;}

[C++] public: \_\_property virtual String\* get\_DataField();public: \_\_property  
virtual void set\_DataField(String\*);

[VB] Overridable Public Property DataField As String

[JScript] public function get DataField() : String;public function set  
DataField(String);

### *Description*

Gets or sets the field name from the data source to bind to the  
**System.Web.UI.WebControls.BoundColumn** .

Use the **System.Web.UI.WebControls.BoundColumn.DataField**  
property to specify the field to bind to the  
**System.Web.UI.WebControls.BoundColumn** .

DataFormatString

ToString

[C#] public virtual string DataFormatString {get; set;}

[C++] public: \_\_property virtual String\* get\_DataFormatString();public:  
\_\_property virtual void set\_DataFormatString(String\*);

[VB] Overridable Public Property DataFormatString As String

[JScript] public function get DataFormatString() : String;public function set  
DataFormatString(String);

1  
2 *Description*

3 Gets or sets the string that specifies the display format for items in the  
4 column.

5 Use the  
6 **System.Web.UI.WebControls.BoundColumn.DataFormatString** property to  
7 provide a custom format for the items in the column.

8 DesignMode

9 FooterStyle

10 FooterText

11 HeaderImageUrl

12 HeaderStyle

13 HeaderText

14 IsTrackingViewState

15 ItemStyle

16 Owner

17 ReadOnly

18 ToString

19  
20  
21 *Description*

22 Gets or sets a value that indicates whether the items in the  
23 **System.Web.UI.WebControls.BoundColumn** can be edited.

24 Use the **System.Web.UI.WebControls.BoundColumn.ReadOnly**  
25 property to specify whether the items in the

**System.Web.UI.WebControls.BoundColumn** can be edited. This property can also be used to programmatically determine whether the column is read-only.

SortExpression

ViewState

Visible

FormatDataValue

[C#] protected virtual string FormatDataValue(object dataValue);

[C++] protected: virtual String\* FormatDataValue(Object\* dataValue);

[VB] Overridable Protected Function FormatDataValue(ByVal dataValue As Object) As String

[JScript] protected function FormatDataValue(dataValue : Object) : String;

#### *Description*

Converts the specified value to the format indicated by the **System.Web.UI.WebControls.BoundColumn.DataFormatString** property.

*Return Value:* The specified value converted to the format indicated by the **System.Web.UI.WebControls.BoundColumn.DataFormatString** property.

Use the **System.Web.UI.WebControls.BoundColumn.FormatDataValue(System.Object)** method to convert the specified value to the format indicated by the **System.Web.UI.WebControls.BoundColumn.DataFormatString** property. The value to format.

Initialize



```

1
2 [C#] public override void Initialize();
3 [C++] public: void Initialize();
4 [VB] Overrides Public Sub Initialize()
5 [JScript] public override function Initialize();
6

```

### *Description*

Resets the **System.Web.UI.WebControls.BoundColumn** to its initial state.

Use the **System.Web.UI.WebControls.BoundColumn.Initialize** method to reset the **System.Web.UI.WebControls.BoundColumn** to its initial state.

### **InitializeCell**

```

14 [C#] public override void InitializeCell(TableCell cell, int columnIndex,
15 ListItemType itemType);
16 [C++] public: void InitializeCell(TableCell* cell, int columnIndex, ListItemType
17 itemType);
18 [VB] Overrides Public Sub InitializeCell(ByVal cell As TableCell, ByVal
19 columnIndex As Integer, ByVal itemType As ListItemType)
20 [JScript] public override function InitializeCell(cell : TableCell, columnIndex : int,
21 itemType : ListItemType);
22

```

### *Description*

Resets the specified cell in the **System.Web.UI.WebControls.BoundColumn** to its initial state.

The **System.Web.UI.WebControls.BoundColumn.InitializeCell(System.Web.UI.WebControls.TableCell,System.Int32,System.Web.UI.WebControls.ListItemType)** method resets the specified cell in the **System.Web.UI.WebControls.BoundColumn** to its initial state. A **System.Web.UI.WebControls.TableCell** object that represents the cell to reset. The column number where the cell is located. One of the **System.Web.UI.WebControls.ListItemType** values.

Button class (System.Web.UI.WebControls)

TrackViewState

### *Description*

Displays a push button control on the Web page.

Use the **System.Web.UI.WebControls.Button** control to create a push button on the Web page. You can create either a **submit** button or a **command** button.

Button

### *Example Syntax:*

TrackViewState

[C#] public Button();

[C++] public: Button();

[VB] Public Sub New()

[JScript] public function Button();

1  
2 *Description*

3       Initializes a new instance of the **System.Web.UI.WebControls.Button**  
4 class.

5       Use this constructor to create and initialize a new instance of the  
6 **System.Web.UI.WebControls.Button** class.

7       AccessKey

8       Attributes

9       BackColor

10      BorderColor

11      BorderStyle

12      BorderWidth

13      CausesValidation

14      TrackViewState

15  
16  
17 *Description*

18       Gets or sets a value indicating whether validation is performed when the  
19 **System.Web.UI.WebControls.Button** control is clicked.

20       By default, page validation is performed when a  
21 **System.Web.UI.WebControls.Button** control is clicked. Page validation  
22 determines whether the input controls associated with a validation control on the  
23 page all pass the validation rules specified by the validation control.

24      ChildControlsCreated

25      ClientID

CommandArgument

TrackViewState

*Description*

Gets or sets an optional parameter passed to the **System.Web.UI.WebControls.Button.Command** event along with the associated **System.Web.UI.WebControls.Button.CommandName** .

Use the **System.Web.UI.WebControls.Button.CommandArgument** property to specify an parameter that compliments the **System.Web.UI.WebControls.Button.CommandName** property.

CommandName

TrackViewState

[C#] public string CommandName {get; set;}

[C++] public: \_\_property String\* get\_CommandName();public: \_\_property void set\_CommandName(String\*);

[VB] Public Property CommandName As String

[JScript] public function get CommandName() : String;public function set CommandName(String);

*Description*

Gets or sets the command name associated with the **System.Web.UI.WebControls.Button** control that is passed to the **System.Web.UI.WebControls.Button.Command** event.

When you have multiple **System.Web.UI.WebControls.Button** controls on a Web page, use the **System.Web.UI.WebControls.Button.CommandName** property to specify or determine the command name associated with each **System.Web.UI.WebControls.Button** control. You can set the **System.Web.UI.WebControls.Button.CommandName** property with any string that identifies the command to perform. You can then programmatically determine the command name of the **System.Web.UI.WebControls.Button** control and perform the appropriate actions.

Context

Controls

ControlStyle

ControlStyleCreated

CssClass

Enabled

EnableViewState

Events

Font

ForeColor

HasChildViewState

Height

ID

IsTrackingViewState

NamingContainer

Page

Parent

1 Site  
2 Style  
3 TabIndex  
4 TagKey  
5 TagName  
6 TemplateSourceDirectory  
7 Text  
8 TrackViewState  
9  
10

11 *Description*

12 Gets or sets the text caption displayed in the  
13 **System.Web.UI.WebControls.Button** control.

14 Use the **System.Web.UI.WebControls.Button.Text** property to specify or  
15 determine the caption to display in the **System.Web.UI.WebControls.Button**  
16 control.

17 ToolTip  
18 UniqueID  
19 ViewState  
20 ViewStateIgnoresCase  
21 Visible  
22 Width  
23 TrackViewState  
24  
25

1  
2  
3 *Description*

4 Occurs when the **System.Web.UI.WebControls.Button** control is clicked.

5 The **System.Web.UI.WebControls.Button.Click** event is raised when the  
6 **System.Web.UI.WebControls.Button** control is clicked. This event is commonly  
7 used when no command name is associated with the  
8 **System.Web.UI.WebControls.Button** control (for instance, with a **submit**  
9 button).

10 **TrackViewState**

11  
12 [C#] public event CommandEventHandler Command;

13 [C++] public: \_\_event CommandEventHandler\* Command;

14 [VB] Public Event Command As CommandEventHandler

15  
16 *Description*

17 Occurs when the **System.Web.UI.WebControls.Button** control is clicked.

18 The **System.Web.UI.WebControls.Button.Command** event is raised  
19 when the **System.Web.UI.WebControls.Button** control is clicked. This event is  
20 commonly used when a command name, such as **Sort** , is associated with the  
21 **System.Web.UI.WebControls.Button** control. This allows you to create multiple  
22 **System.Web.UI.WebControls.Button** controls on a Web page and  
23 programmatically determine which **System.Web.UI.WebControls.Button** control  
24 is clicked.

25 **AddAttributesToRender**

1  
2 [C#] protected override void AddAttributesToRender(HtmlTextWriter writer);

3 [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);

4 [VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As  
5 HtmlTextWriter)

6 [JScript] protected override function AddAttributesToRender(writer :  
7 HtmlTextWriter);

8  
9 *Description*

10 Adds the attributes of the **System.Web.UI.WebControls.Button** control to  
11 the output stream for rendering on the client. An  
12 **System.Web.UI.HtmlTextWriter** that contains the output stream to render on the  
13 client.

14 **OnClick**

15  
16 [C#] protected virtual void OnClick(EventArgs e);

17 [C++] protected: virtual void OnClick(EventArgs\* e);

18 [VB] Overridable Protected Sub OnClick(ByVal e As EventArgs)

19 [JScript] protected function OnClick(e : EventArgs);

20  
21 *Description*

22 Raises the **System.Web.UI.WebControls.Button.Click** event of the  
23 **System.Web.UI.WebControls.Button** control.

24 The **System.Web.UI.WebControls.Button.Click** event is raised when the  
25 **System.Web.UI.WebControls.Button** control is clicked. This event is commonly



used when no command name is associated with the

**System.Web.UI.WebControls.Button** control (for instance, with a **submit** button). A **System.EventArgs** that contains the event data.

#### OnCommand

[C#] protected virtual void OnCommand(CommandEventArgs e);

[C++] protected: virtual void OnCommand(CommandEventArgs\* e);

[VB] Overridable Protected Sub OnCommand(ByVal e As CommandEventArgs)

[JScript] protected function OnCommand(e : CommandEventArgs);

#### *Description*

Raises the **System.Web.UI.WebControls.Button.Command** event of the **System.Web.UI.WebControls.Button** control.

The **System.Web.UI.WebControls.Button.Command** event is raised when the **System.Web.UI.WebControls.Button** control is clicked. This event is commonly used when a command name, such as **Sort**, is associated with the **System.Web.UI.WebControls.Button** control. This allows you to create multiple **System.Web.UI.WebControls.Button** controls on a Web page and programmatically determine which **System.Web.UI.WebControls.Button** control is clicked. A **System.Web.UI.WebControls.CommandEventArgs** that contains the event data.

#### RenderContents

[C#] protected override void RenderContents(HtmlTextWriter writer);

[C++] protected: void RenderContents(HtmlTextWriter\* writer);

1 [VB] Overrides Protected Sub RenderContents(ByVal writer As HtmlTextWriter)

2 [JScript] protected override function RenderContents(writer : HtmlTextWriter);

3  
4 *Description*

5 IPostBackEventHandler.RaisePostBackEvent

6  
7 [C#] void IPostBackEventHandler.RaisePostBackEvent(string eventArgument);

8 [C++] void IPostBackEventHandler::RaisePostBackEvent(String\*  
9 eventArgument);

10 [VB] Sub RaisePostBackEvent(ByVal eventArgument As String) Implements

11 IPostBackEventHandler.RaisePostBackEvent

12 [JScript] function IPostBackEventHandler.RaisePostBackEvent(eventArgument :  
13 String);

14 ButtonColumn class (System.Web.UI.WebControls)

15 TrackViewState

16  
17  
18 *Description*

19 A column type for the **System.Web.UI.WebControls.DataGrid** control  
20 that contains a user-defined command button, such as **Add** or **Remove** , that  
21 corresponds with each row in the column.

22 Use the **System.Web.UI.WebControls.ButtonColumn** column type in a  
23 **System.Web.UI.WebControls.DataGrid** control to create a command button that  
24 corresponds with each row in the **System.Web.UI.WebControls.DataGrid**

control. Specify the caption displayed in the command buttons by setting the **System.Web.UI.WebControls.ButtonColumn.Text** property.

ButtonColumn

*Example Syntax:*

TrackViewState

[C#] public ButtonColumn();

[C++] public: ButtonColumn();

[VB] Public Sub New()

[JScript] public function ButtonColumn();

#### *Description*

Initializes a new instance of the **System.Web.UI.WebControls.ButtonColumn** class.

Use this constructor to create and initialize a new instance of the **System.Web.UI.WebControls.ButtonColumn** class.

ButtonType

TrackViewState

[C#] public virtual ButtonColumnType ButtonType {get; set;}

[C++] public: \_\_property virtual ButtonColumnType get\_ButtonType();public:

\_\_property virtual void set\_ButtonType(ButtonColumnType);

[VB] Overridable Public Property ButtonType As ButtonColumnType

[JScript] public function get ButtonType() : ButtonColumnType;public function

set ButtonType(ButtonColumnType);

1  
2 *Description*

3 Gets or sets the type of button to display in the  
4 **System.Web.UI.WebControls.ButtonColumn** column.

5 Use this property to specify whether the buttons in the column are  
6 displayed as link or push buttons. Clicking on the command buttons in the column  
7 raises the **System.Web.UI.WebControls.DataGrid.ItemCommand** event. You  
8 can programmatically control the action performed when the command button is  
9 clicked by providing an event handler for the

10 **System.Web.UI.WebControls.DataGrid.ItemCommand** event.

11 CommandName

12 TrackViewState

13  
14 [C#] public virtual string CommandName {get; set;}

15 [C++] public: \_\_property virtual String\* get\_CommandName();public: \_\_property  
16 virtual void set\_CommandName(String\*);

17 [VB] Overridable Public Property CommandName As String

18 [JScript] public function get CommandName() : String;public function set  
19 CommandName(String);

20  
21 *Description*

22 Gets or sets a string that represents the command to perform when a button  
23 in the **System.Web.UI.WebControls.ButtonColumn** is clicked.

24 Use the **System.Web.UI.WebControls.ButtonColumn.CommandName**  
25 property to associate a command name, such as **Add** or **Remove** , with a button.

You can set the **System.Web.UI.WebControls.ButtonColumn.CommandName** property to any string that identifies the action to perform when the command button is clicked. You can then programmatically determine the command name in the event handler for the **System.Web.UI.WebControls.DataGrid.ItemCommand** event and perform the appropriate actions.

DataTextField

TrackViewState

[C#] public virtual string DataTextField {get; set;}

[C++] public: \_\_property virtual String\* get\_DataTextField();public: \_\_property virtual void set\_DataTextField(String\*);

[VB] Overridable Public Property DataTextField As String

[JScript] public function get DataTextField() : String;public function set DataTextField(String);

### *Description*

Gets or sets the field name from a data source to bind to the

**System.Web.UI.WebControls.ButtonColumn** .

Use the **System.Web.UI.WebControls.ButtonColumn.DataTextField** property to specify the field name from the data source to bind to the **System.Web.UI.WebControls.ButtonColumn.Text** property of the buttons in the **System.Web.UI.WebControls.ButtonColumn** .

DataTextFormatString

TrackViewState

```

1
2 [C#] public virtual string DataTextFormatString {get; set;}
3 [C++] public: __property virtual String* get_DataTextFormatString();public:
4 __property virtual void set_DataTextFormatString(String*);
5 [VB] Overridable Public Property DataTextFormatString As String
6 [JScript] public function get DataTextFormatString() : String;public function set
7 DataTextFormatString(String);
8

```

### *Description*

Gets or sets the string that specifies the display format for the caption in each command button.

Use the **System.Web.UI.WebControls.ButtonColumn.DataTextFormatString** property to provide a custom display format for the caption of the command buttons in the column.

DesignMode

FooterStyle

FooterText

HeaderImageUrl

HeaderStyle

HeaderText

IsTrackingViewState

ItemStyle

Owner

SortExpression

1       Text  
2       TrackViewState

3  
4  
5    *Description*

6       Gets or sets the caption displayed in the command buttons of the

7    **System.Web.UI.WebControls.ButtonColumn .**

8       Use the **System.Web.UI.WebControls.ButtonColumn.Text** property to  
9    specify or determine the caption displayed in the command buttons of the

10   **System.Web.UI.WebControls.ButtonColumn .**

11       ViewState

12       Visible

13       FormatDataTextValue

14  
15    [C#] protected virtual string FormatDataTextValue(object dataTextValue);

16    [C++] protected: virtual String\* FormatDataTextValue(Object\* dataTextValue);

17    [VB] Overridable Protected Function FormatDataTextValue(ByVal

18    dataTextValue As Object) As String

19    [JScript] protected function FormatDataTextValue(dataTextValue : Object) :

20    String;

21  
22    *Description*

23       Converts the specified value to the format indicated by the

24    **System.Web.UI.WebControls.ButtonColumn.DataTextFormatString**

25    property.

*Return Value:* The specified value converted to the format indicated by the **System.Web.UI.WebControls.ButtonColumn.DataTextFormatString** property.

Use the **System.Web.UI.WebControls.ButtonColumn.FormatDataTextValue(System.Object)** method to convert the specified value to the format indicated by the **System.Web.UI.WebControls.ButtonColumn.DataTextFormatString** property. The value to format.

#### Initialize

[C#] public override void Initialize();

[C++] public: void Initialize();

[VB] Overrides Public Sub Initialize()

[JScript] public override function Initialize();

#### Description

Resets the **System.Web.UI.WebControls.ButtonColumn** to its initial state.

Use the **System.Web.UI.WebControls.ButtonColumn.Initialize** method to reset the **System.Web.UI.WebControls.ButtonColumn** to its initial state.

#### InitializeCell

[C#] public override void InitializeCell(TableCell cell, int columnIndex, ListItemType itemType);

[C++] public: void InitializeCell(TableCell\* cell, int columnIndex, ListItemType



```

1 itemType);
2 [VB] Overrides Public Sub InitializeCell(ByVal cell As TableCell, ByVal
3 columnIndex As Integer, ByVal itemType As ListItemType)
4 [JScript] public override function InitializeCell(cell : TableCell, columnIndex : int,
5 itemType : ListItemType);
6

```

### *Description*

Initializes a cell in the **System.Web.UI.WebControls.ButtonColumn** to its initial state.

The **System.Web.UI.WebControls.ButtonColumn.InitializeCell(System.Web.UI.WebControls.TableCell, System.Int32, System.Web.UI.WebControls.ListItemType)** method resets the specified cell in the **System.Web.UI.WebControls.ButtonColumn** to its initial state. A **System.Web.UI.WebControls.TableCell** object that represents the cell to reset. The column number where the cell is located. One of the **System.Web.UI.WebControls.ListItemType** values.

ButtonColumnType enumeration (System.Web.UI.WebControls)

TrackViewState

### *Description*

Specifies the button type for the **System.Web.UI.WebControls.ButtonColumn** object.

1       The **System.Web.UI.WebControls.ButtonColumnType** enumeration  
2 represents the button styles for the  
3 **System.Web.UI.WebControls.ButtonColumn** object.

4       TrackViewState

5  
6   [C#] public const ButtonColumnType LinkButton;  
7   [C++] public: const ButtonColumnType LinkButton;  
8   [VB] Public Const LinkButton As ButtonColumnType  
9   [JScript] public var LinkButton : ButtonColumnType;

10  
11   *Description*

12       A column of hyperlink style buttons.

13       TrackViewState

14  
15   [C#] public const ButtonColumnType PushButton;  
16   [C++] public: const ButtonColumnType PushButton;  
17   [VB] Public Const PushButton As ButtonColumnType  
18   [JScript] public var PushButton : ButtonColumnType;

19  
20   *Description*

21       A column of push buttons.

22       Calendar class (System.Web.UI.WebControls)

23       ToString

1  
2  
3 *Description*

4       Displays a single month calendar that allows the user to select dates and  
5 move to the next or previous month.

6       Use the **System.Web.UI.WebControls.Calendar** control to display a  
7 single month of a calendar on a Web page. The control allows you to select dates  
8 and move to the next or previous month. The

9 **System.Web.UI.WebControls.Calendar** control supports all of the

10 **System.Globalization.Calendar** types in the **System.Globalization** namespace.

11 Apart from the Gregorian calendar, this also includes calendars that use different  
12 year and month systems, such as the Hjiri calendar.

13       Calendar

14       *Example Syntax:*

15       ToString

16  
17 [C#] public Calendar();

18 [C++] public: Calendar();

19 [VB] Public Sub New()

20 [JScript] public function Calendar();

21  
22 *Description*

23       Initializes a new instance of the **System.Web.UI.WebControls.Calendar**  
24 class.

1 Use this constructor to create and initialize a new instance of the  
2 **System.Web.UI.WebControls.Calendar** class.

3 AccessKey

4 Attributes

5 BackColor

6 BorderColor

7 BorderStyle

8 BorderWidth

9 CellPadding

10 ToString

11  
12  
13 *Description*

14 Gets or sets the amount of space between the contents of a cell and the  
15 cell's border.

16 Use this property to control the spacing between the contents of a cell and  
17 the cell's border. The padding amount specified is added to all four sides of a cell.  
18 Individual cell sizes cannot be specified.

19 CellSpacing

20 ToString

21  
22 [C#] public int CellSpacing {get; set;}

23 [C++] public: \_\_property int get\_CellSpacing();public: \_\_property void  
24 set\_CellSpacing(int);

25 [VB] Public Property CellSpacing As Integer

1 [JScript] public function get CellSpacing() : int;public function set  
2 CellSpacing(int);

3  
4 *Description*

5 Gets or sets the amount of space between cells.  
6 Use this property to control the spacing between individual cells in the  
7 calendar. This spacing is applied both vertically and horizontally.

8 ChildControlsCreated

9 ClientID

10 Context

11 Controls

12 ControlStyle

13 ControlStyleCreated

14 CssClass

15 DayHeaderStyle

16 ToString

17  
18  
19 *Description*

20 Gets the style properties for the section that displays the day of the week.  
21 Use this property to specify the style for the section that displays the days  
22 of the week. For additional information on the different style properties that can be  
23 controlled, see **System.Web.UI.WebControls.TableItemStyle** .

24 DayNameFormat

25 ToString

```

1  [C#] public DayNameFormat DayNameFormat {get; set;}
2
3  [C++] public: __property DayNameFormat get _DayNameFormat();public:
4  __property void set _DayNameFormat(DayNameFormat);
5
6  [VB] Public Property DayNameFormat As DayNameFormat
7
8  [JScript] public function get DayNameFormat() : DayNameFormat;public function
9  set DayNameFormat(DayNameFormat);
10

```

### *Description*

Gets or sets the name format of days of the week.

Use the **System.Web.UI.WebControls.Calendar.DayNameFormat** property to specify the name format for the days of the week. This property is set with one of the **System.Web.UI.WebControls.DayNameFormat** enumeration values. You can specify whether the days of the week are displayed as the full name, short (abbreviated) name, first letter of the day, or first two letters of the day.

DayStyle

ToString

```

20 [C#] public TableItemStyle DayStyle {get;}
21
22 [C++] public: __property TableItemStyle* get _DayStyle();
23
24 [VB] Public ReadOnly Property DayStyle As TableItemStyle
25
26 [JScript] public function get DayStyle() : TableItemStyle;

```

### *Description*

Gets the style properties for the days in the displayed month.

Use this property to specify the style for the days in the displayed month.

Enabled

EnableViewState

Events

FirstDayOfWeek

ToString

### *Description*

Gets or sets the day of the week to display in the first day column of the **System.Web.UI.WebControls.Calendar** control.

Use the **System.Web.UI.WebControls.Calendar.FirstDayOfWeek** property to specify the day of the week to display in the first day column of the **System.Web.UI.WebControls.Calendar** control. This property is set with one of the **System.Web.UI.WebControls.FirstDayOfWeek** enumeration values. You can specify any day of the week or **FirstDayOfWeek.Default**, which indicates that the date is determined by the system settings.

Font

ForeColor

HasChildViewState

Height

ID

IsTrackingViewState

NamingContainer

NextMonthText

ToString

*Description*

Gets or sets the text displayed for the next month navigation control.

Use the **System.Web.UI.WebControls.Calendar.NextMonthText** property to provide custom text for the next month navigation control. This property is commonly used along with the **System.Web.UI.WebControls.Calendar.PrevMonthText** property to provide a custom set of navigation controls.

NextPrevFormat

ToString

[C#] public NextPrevFormat NextPrevFormat {get; set;}

[C++] public: \_\_property NextPrevFormat get\_NextPrevFormat();public:

\_\_property void set\_NextPrevFormat(NextPrevFormat);

[VB] Public Property NextPrevFormat As NextPrevFormat

[JScript] public function get NextPrevFormat() : NextPrevFormat;public function set NextPrevFormat(NextPrevFormat);

*Description*

Gets or sets the format of the next and previous month navigation elements in the title section of the **System.Web.UI.WebControls.Calendar** control.



1 Use the **System.Web.UI.WebControls.Calendar.NextPrevFormat**  
2 property to specify the format for the next and previous month navigation  
3 elements on the **System.Web.UI.WebControls.Calendar** control. This property  
4 is set with one of the **System.Web.UI.WebControls.NextPrevFormat**  
5 enumeration values. You can specify whether the text for the navigation controls  
6 is displayed as the full month name, the short (abbreviated) month name, or  
7 custom text.

8 NextPrevStyle

9 ToString

10  
11 [C#] public TableItemStyle NextPrevStyle {get;}

12 [C++] public: \_\_property TableItemStyle\* get\_NextPrevStyle();

13 [VB] Public ReadOnly Property NextPrevStyle As TableItemStyle

14 [JScript] public function get NextPrevStyle() : TableItemStyle;

15  
16 *Description*

17 Gets the style properties for the next and previous month navigation  
18 elements.

19 Use this property to specify the style for the next and previous month  
20 navigation elements. For additional information on the different style properties  
21 that can be controlled, see **System.Web.UI.WebControls.TableItemStyle** .

22 OtherMonthDayStyle

23 ToString

24  
25 [C#] public TableItemStyle OtherMonthDayStyle {get;}

1 [C++] public: \_\_property TableItemStyle\* get\_OtherMonthDayStyle();  
2 [VB] Public ReadOnly Property OtherMonthDayStyle As TableItemStyle  
3 [JScript] public function get OtherMonthDayStyle() : TableItemStyle;

#### 4 5 *Description*

6 Gets the style properties for the days on the  
7 **System.Web.UI.WebControls.Calendar** control that are not in the displayed  
8 month.

9 Use this property to specify the style for the days on the  
10 **System.Web.UI.WebControls.Calendar** control that are not in the displayed  
11 month. If the **System.Web.UI.WebControls.Calendar.OtherMonthDayStyle**  
12 property is not set, the style specified in the  
13 **System.Web.UI.WebControls.Calendar.DayStyle** property is used.

14 Page

15 Parent

16 PrevMonthText

17 ToString

#### 18 19 20 *Description*

21 Gets or sets the text displayed for the previous month navigation control.

22 Use the **System.Web.UI.WebControls.Calendar.PrevMonthText**  
23 property to provide custom text for the previous month navigation control. This  
24 property is commonly used along with the  
25

**System.Web.UI.WebControls.Calendar.NextMonthText** property to provide a custom set of navigation controls.

**SelectedDate**

**ToString**

[C#] public DateTime SelectedDate {get; set;}

[C++] public: \_\_property DateTime get\_SelectedDate();public: \_\_property void set\_SelectedDate(DateTime);

[VB] Public Property SelectedDate As DateTime

[JScript] public function get SelectedDate() : DateTime;public function set SelectedDate(DateTime);

### *Description*

Gets or sets the selected date.

Use the **System.Web.UI.WebControls.Calendar.SelectedDate** property to determine the selected date on the **System.Web.UI.WebControls.Calendar** control. The **System.Web.UI.WebControls.Calendar.SelectedDate** property is typically used when the **System.Web.UI.WebControls.Calendar.SelectionMode** property is set to **CalendarSelectionMode.Day** . This setting only allows a single date selection from the **System.Web.UI.WebControls.Calendar** control.

**SelectedDates**

**ToString**

[C#] public SelectedDatesCollection SelectedDates {get;}

[C++] public: \_\_property SelectedDatesCollection\* get\_SelectedDates();

1 [VB] Public ReadOnly Property SelectedDates As SelectedDatesCollection

2 [JScript] public function get SelectedDates() : SelectedDatesCollection;

3  
4 *Description*

5 Gets a collection of **System.DateTime** objects that represent the selected  
6 dates on the **System.Web.UI.WebControls.Calendar** control.

7 Use the **System.Web.UI.WebControls.Calendar.SelectedDates**  
8 collection to determine the currently selected dates on the  
9 **System.Web.UI.WebControls.Calendar** control. This property is typically used  
10 when the **System.Web.UI.WebControls.Calendar.SelectionMode** property is set  
11 to **CalendarSelectionMode.DayWeek** or  
12 **CalendarSelectionMode.DayWeekMonth** . These settings allow you to select  
13 multiple dates from the **System.Web.UI.WebControls.Calendar** control by week  
14 or month.

15 SelectedDayStyle

16 ToString

17  
18 [C#] public TableItemStyle SelectedDayStyle {get;}

19 [C++] public: \_\_property TableItemStyle\* get\_SelectedDayStyle();

20 [VB] Public ReadOnly Property SelectedDayStyle As TableItemStyle

21 [JScript] public function get SelectedDayStyle() : TableItemStyle;

22  
23 *Description*

24 Gets the style properties for the selected dates.

Use this property to specify the style for the selected dates on the **System.Web.UI.WebControls.Calendar** control. If the **System.Web.UI.WebControls.Calendar.SelectedDayStyle** property is not set, the style specified in the **System.Web.UI.WebControls.Calendar.DayStyle** property is used.

SelectionMode

ToString

[C#] public CalendarSelectionMode SelectionMode {get; set;}

[C++] public: \_\_property CalendarSelectionMode get\_SelectionMode();public:

\_\_property void set\_SelectionMode(CalendarSelectionMode);

[VB] Public Property SelectionMode As CalendarSelectionMode

[JScript] public function get SelectionMode() : CalendarSelectionMode;public

function set SelectionMode(CalendarSelectionMode);

### *Description*

Gets or sets the date selection mode on the **System.Web.UI.WebControls.Calendar** control that specifies whether the user can select a single day, a week, or an entire month.

Use the **System.Web.UI.WebControls.Calendar.SelectionMode** property to specify the date selection mode on the **System.Web.UI.WebControls.Calendar** control. This property is set with one of the **System.Web.UI.WebControls.CalendarSelectionMode** enumeration values. You can specify whether the user can select a single day, a week, or an entire month, or you can disable date selection capability entirely.

SelectMonthText

ToString

[C#] public string SelectMonthText {get; set;}

[C++] public: \_\_property String\* get\_SelectMonthText();public: \_\_property void  
set\_SelectMonthText(String\*);

[VB] Public Property SelectMonthText As String

[JScript] public function get SelectMonthText() : String;public function set  
SelectMonthText(String);

#### *Description*

Gets or sets the text displayed for the month selection element in the  
selector column.

Use the **System.Web.UI.WebControls.Calendar.SelectMonthText**  
property to provide custom text for the month selection element in the selector  
column.

SelectorStyle

ToString

[C#] public TableItemStyle SelectorStyle {get;}

[C++] public: \_\_property TableItemStyle\* get\_SelectorStyle();

[VB] Public ReadOnly Property SelectorStyle As TableItemStyle

[JScript] public function get SelectorStyle() : TableItemStyle;

#### *Description*

1 Gets the style properties for the week and month selector column.

2 Use this property to specify the style for the week and month selector  
3 column. For additional information on the different style properties that can be  
4 controlled, see **System.Web.UI.WebControls.TableItemStyle** .

5 SelectWeekText

6 ToString

7  
8 [C#] public string SelectWeekText {get; set;}

9 [C++] public: \_\_property String\* get\_SelectWeekText();public: \_\_property void  
10 set\_SelectWeekText(String\*);

11 [VB] Public Property SelectWeekText As String

12 [JScript] public function get SelectWeekText() : String;public function set  
13 SelectWeekText(String);

14  
15 *Description*

16 Gets or sets the text displayed for the week selection element in the selector  
17 column.

18 Use the **System.Web.UI.WebControls.Calendar.SelectWeekText**  
19 property to provide custom text for the week selection element in the selector  
20 column.

21 ShowDayHeader

22 ToString

23  
24 [C#] public bool ShowDayHeader {get; set;}

25 [C++] public: \_\_property bool get\_ShowDayHeader();public: \_\_property void

1 set\_ShowDayHeader(bool);

2 [VB] Public Property ShowDayHeader As Boolean

3 [JScript] public function get ShowDayHeader() : Boolean;public function set

4 ShowDayHeader(Boolean);

5  
6 *Description*

7 Gets or sets a value indicating whether the heading for the days of the week  
8 is displayed.

9 Use the **System.Web.UI.WebControls.Calendar.ShowDayHeader**  
10 property to specify whether the heading for the days of the week is displayed.

11 ShowGridLines

12 ToString

13  
14 [C#] public bool ShowGridLines {get; set;}

15 [C++] public: \_\_property bool get\_ShowGridLines();public: \_\_property void

16 set\_ShowGridLines(bool);

17 [VB] Public Property ShowGridLines As Boolean

18 [JScript] public function get ShowGridLines() : Boolean;public function set

19 ShowGridLines(Boolean);

20  
21 *Description*

22 Gets or sets a value indicating whether the days on the  
23 **System.Web.UI.WebControls.Calendar** control are separated with grid lines.



1        Use the **System.Web.UI.WebControls.Calendar.ShowGridLines**  
2        property to specify whether the days on the  
3        **System.Web.UI.WebControls.Calendar** control are separated with grid lines.

4        ShowNextPrevMonth

5        ToString

7        [C#] public bool ShowNextPrevMonth {get; set;}

8        [C++] public: \_\_property bool get\_ShowNextPrevMonth();public: \_\_property  
9        void set\_ShowNextPrevMonth(bool);

10       [VB] Public Property ShowNextPrevMonth As Boolean

11       [JScript] public function get ShowNextPrevMonth() : Boolean;public function set  
12       ShowNextPrevMonth(Boolean);

14       *Description*

15       Gets or sets a value indicating whether the  
16       **System.Web.UI.WebControls.Calendar** control displays the next and previous  
17       month navigation elements in the title section.

18       Use the **System.Web.UI.WebControls.Calendar.ShowNextPrevMonth**  
19       property to specify whether the next and previous month navigation elements are  
20       displayed in the title section.

21       ShowTitle

22       ToString

24       [C#] public bool ShowTitle {get; set;}

25       [C++] public: \_\_property bool get\_ShowTitle();public: \_\_property void

1 set\_ShowTitle(bool);

2 [VB] Public Property ShowTitle As Boolean

3 [JScript] public function get ShowTitle() : Boolean;public function set

4 ShowTitle(Boolean);

6 *Description*

7 Gets or sets a value indicating whether the title section is displayed.

8 Use the **System.Web.UI.WebControls.Calendar.ShowTitle** property to  
9 specify whether the title section is displayed.

10 Site

11 Style

12 TabIndex

13 TagKey

14 TagName

15 TemplateSourceDirectory

16 TitleFormat

17 ToString

20 *Description*

21 Gets or sets the title format for the title section.

22 Use the **System.Web.UI.WebControls.Calendar.TitleFormat** property to  
23 specify the format for the title section. This property is set with one of the  
24 **System.Web.UI.WebControls.TitleFormat** enumeration values. You can specify  
25 whether the title displays as the month or both the month and the year.

1 TitleStyle  
 2 ToString  
 3  
 4 [C#] public TableItemStyle TitleStyle {get;}  
 5 [C++] public: \_\_property TableItemStyle\* get\_TitleStyle();  
 6 [VB] Public ReadOnly Property TitleStyle As TableItemStyle  
 7 [JScript] public function get TitleStyle() : TableItemStyle;

8  
 9 *Description*

10 Gets the style properties of the title heading for the  
 11 **System.Web.UI.WebControls.Calendar** control.

12 Use this property to specify the style for the title heading of the  
 13 **System.Web.UI.WebControls.Calendar** control. For additional information on  
 14 the different style properties that can be controlled, see  
 15 **System.Web.UI.WebControls.TableItemStyle** .

16 TodayDayStyle  
 17 ToString  
 18

19 [C#] public TableItemStyle TodayDayStyle {get;}  
 20 [C++] public: \_\_property TableItemStyle\* get\_TodayDayStyle();  
 21 [VB] Public ReadOnly Property TodayDayStyle As TableItemStyle  
 22 [JScript] public function get TodayDayStyle() : TableItemStyle;

23  
 24 *Description*  
 25

1 Gets the style properties for today's date on the  
2 **System.Web.UI.WebControls.Calendar** control.

3 Use this property to specify the style for today's date on the  
4 **System.Web.UI.WebControls.Calendar** control. If the  
5 **System.Web.UI.WebControls.Calendar.TodayDayStyle** property is not set, the  
6 style specified in the **System.Web.UI.WebControls.Calendar.DayStyle** property  
7 is used.

8 **TodaysDate**

9 **ToString**

10  
11 [C#] public DateTime TodaysDate {get; set;}

12 [C++] public: \_\_property DateTime get\_TodaysDate();public: \_\_property void  
13 set\_TodaysDate(DateTime);

14 [VB] Public Property TodaysDate As DateTime

15 [JScript] public function get TodaysDate() : DateTime;public function set  
16 TodaysDate(DateTime);

17  
18 *Description*

19 Gets or sets the value for today's date.

20 Use the **System.Web.UI.WebControls.Calendar.TodaysDate** property to  
21 determine today's date. You can also use this property to programmatically set the  
22 value for today's date on the **System.Web.UI.WebControls.Calendar** control.

23 This property is set using a **System.DateTime** object.

24 **ToolTip**

25 **UniqueID**

ViewState

ViewStateIgnoresCase

Visible

VisibleDate

ToString

*Description*

Gets or sets the date that specifies the month to display on the **System.Web.UI.WebControls.Calendar** control.

Use the **System.Web.UI.WebControls.Calendar.VisibleDate** property to programmatically set the month to display on the **System.Web.UI.WebControls.Calendar** control. The specified date can be any date in the month you want to display, although it is common to use the first date in the month. This property is set using a **System.DateTime** object.

WeekendDayStyle

ToString

[C#] public TableItemStyle WeekendDayStyle {get;}

[C++] public: \_\_property TableItemStyle\* get\_WeekendDayStyle();

[VB] Public ReadOnly Property WeekendDayStyle As TableItemStyle

[JScript] public function get WeekendDayStyle() : TableItemStyle;

*Description*

1 Gets the style properties for the weekend dates on the  
2 **System.Web.UI.WebControls.Calendar** control.

3 Use this property to specify the style for the weekend dates on the  
4 **System.Web.UI.WebControls.Calendar** control. If the  
5 **System.Web.UI.WebControls.Calendar.WeekendDayStyle** property is not set,  
6 the style specified in the **System.Web.UI.WebControls.Calendar.DayStyle**  
7 property is used.

8 Width

9 ToString

10  
11  
12 *Description*

13 Occurs when each day is created in the control hierarchy for the  
14 **System.Web.UI.WebControls.Calendar** control.

15 This event is raised when each day is created in the control hierarchy for  
16 the **System.Web.UI.WebControls.Calendar** control.

17 ToString

18  
19  
20 *Description*

21 Occurs when the user selects a day, a week, or an entire month by clicking  
22 the date selector controls.

23 This event is raised when the user selects a day, a week, or an entire month  
24 by clicking the date selector controls.

25 ToString

1  
2  
3 *Description*

4 Occurs when the user clicks on the next or previous month navigation  
5 controls on the title heading.

6 This event is raised when the user clicks on the next or previous month  
7 navigation elements on the title heading.

8 CreateControlCollection

9  
10 [C#] protected override ControlCollection CreateControlCollection();

11 [C++] protected: ControlCollection\* CreateControlCollection();

12 [VB] Overrides Protected Function CreateControlCollection() As

13 ControlCollection

14 [JScript] protected override function CreateControlCollection() :

15 ControlCollection;

16  
17 *Description*

18  
19 HasWeekSelectors

20  
21 [C#] protected bool HasWeekSelectors(CalendarSelectionMode selectionMode);

22 [C++] protected: bool HasWeekSelectors(CalendarSelectionMode  
23 selectionMode);

24 [VB] Protected Function HasWeekSelectors(ByVal selectionMode As

25 CalendarSelectionMode) As Boolean

1 [JScript] protected function HasWeekSelectors(selectionMode :  
2 CalendarSelectionMode) : Boolean;

3  
4 *Description*

5 Determines if a **System.Web.UI.WebControls.CalendarSelectionMode**  
6 contains week selectors.

7 *Return Value:* **true** if the

8 **System.Web.UI.WebControls.CalendarSelectionMode** contains week selectors;  
9 otherwise **false** .

10 **System.Web.UI.WebControls.CalendarSelectionMode** values of  
11 **DayWeek** and **DayWeekMonth** contains week selectors. One of the  
12 **System.Web.UI.WebControls.CalendarSelectionMode** values.

13 **LoadViewState**

14  
15 [C#] protected override void LoadViewState(object savedState);

16 [C++] protected: void LoadViewState(Object\* savedState);

17 [VB] Overrides Protected Sub LoadViewState(ByVal savedState As Object)

18 [JScript] protected override function LoadViewState(savedState : Object);

19  
20 *Description*

21 Loads a saved state of the **System.Web.UI.WebControls.Calendar** . An  
22 **System.Object** that contains the saved condition of the  
23 **System.Web.UI.WebControls.Calendar**.

24 **OnDayRender**



1  
2 [C#] protected virtual void OnDayRender(TableCell cell, CalendarDay day);  
3 [C++] protected: virtual void OnDayRender(TableCell\* cell, CalendarDay\* day);  
4 [VB] Overridable Protected Sub OnDayRender(ByVal cell As TableCell, ByVal  
5 day As CalendarDay)  
6 [JScript] protected function OnDayRender(cell : TableCell, day : CalendarDay);  
7

### 8 *Description*

9       Raises the **System.Web.UI.WebControls.Calendar.DayRender** event of  
10 the **System.Web.UI.WebControls.Calendar** control and allows you to provide a  
11 custom handler for the **System.Web.UI.WebControls.Calendar.DayRender**  
12 event.

13       Although databinding is not supported for the  
14 **System.Web.UI.WebControls.Calendar** control, it is possible modify the  
15 content and formatting of the individual date cells. Before the  
16 **System.Web.UI.WebControls.Calendar** control is displayed on the Web page, it  
17 creates and assembles the components that make up the control. The  
18 **System.Web.UI.WebControls.Calendar.DayRender** event is raised when each  
19 date cell in **System.Web.UI.WebControls.Calendar** control is created. You can  
20 control the contents and formatting of a date cell when it is created by providing  
21 code in the event handler for the  
22 **System.Web.UI.WebControls.Calendar.DayRender** event. A  
23 **System.Web.UI.WebControls.TableCell** that contains information about the cell  
24 to render. A **System.Web.UI.WebControls.CalendarDay** that contains  
25 information about the day to render.

## OnSelectionChanged

[C#] protected virtual void OnSelectionChanged();  
[C++] protected: virtual void OnSelectionChanged();  
[VB] Overridable Protected Sub OnSelectionChanged()  
[JScript] protected function OnSelectionChanged();

### *Description*

Raises the **System.Web.UI.WebControls.Calendar.SelectionChanged** event of the **System.Web.UI.WebControls.Calendar** control and allows you to provide a custom handler for the **System.Web.UI.WebControls.Calendar.SelectionChanged** event.

The **System.Web.UI.WebControls.Calendar.SelectionChanged** event is raised when the user selects a day, a week, or an entire month by clicking the date selector controls.

## OnVisibleMonthChanged

[C#] protected virtual void OnVisibleMonthChanged(DateTime newDate, DateTime previousDate);  
[C++] protected: virtual void OnVisibleMonthChanged(DateTime newDate, DateTime previousDate);  
[VB] Overridable Protected Sub OnVisibleMonthChanged(ByVal newDate As DateTime, ByVal previousDate As DateTime)  
[JScript] protected function OnVisibleMonthChanged(newDate : DateTime, previousDate : DateTime);

1  
2 *Description*

3       Raises the  
4 **System.Web.UI.WebControls.Calendar.VisibleMonthChanged** event of the  
5 **System.Web.UI.WebControls.Calendar** control and allows you to provide a  
6 custom handler for the  
7 **System.Web.UI.WebControls.Calendar.VisibleMonthChanged** event.

8       The **System.Web.UI.WebControls.Calendar.VisibleMonthChanged**  
9 event is raised when the user clicks on the next or previous month navigation  
10 elements on the title section. A **System.DateTime** object that represents the month  
11 currently displayed in the **System.Web.UI.WebControls.Calendar** control. A  
12 **System.DateTime** object that represents the previous month displayed by the  
13 **System.Web.UI.WebControls.Calendar** control.

14       Render

15  
16 [C#] protected override void Render(HtmlTextWriter writer);

17 [C++] protected: void Render(HtmlTextWriter\* writer);

18 [VB] Overrides Protected Sub Render(ByVal writer As HtmlTextWriter)

19 [JScript] protected override function Render(writer : HtmlTextWriter);  
20

21 *Description*

22       Displays the **System.Web.UI.WebControls.Calendar** control on the  
23 client. An **System.Web.UI.HtmlTextWriter** that contains the output stream for  
24 rendering on the client.

25       SaveViewState

1  
2 [C#] protected override object SaveViewState();

3 [C++] protected: Object\* SaveViewState();

4 [VB] Overrides Protected Function SaveViewState() As Object

5 [JScript] protected override function SaveViewState() : Object;

6  
7 *Description*

8 Stores the state of the **System.Web.UI.WebControls.Calendar** .

9 *Return Value:* An object that contains the saved state of the

10 **System.Web.UI.WebControls.Calendar** .

11 **IPostBackEventHandler.RaisePostBackEvent**

12  
13 [C#] void IPostBackEventHandler.RaisePostBackEvent(string eventArgument);

14 [C++] void IPostBackEventHandler::RaisePostBackEvent(String\*  
15 eventArgument);

16 [VB] Sub RaisePostBackEvent(ByVal eventArgument As String) Implements

17 IPostBackEventHandler.RaisePostBackEvent

18 [JScript] function IPostBackEventHandler.RaisePostBackEvent(eventArgument :  
19 String);

20 **TrackViewState**

21  
22 [C#] protected override void TrackViewState();

23 [C++] protected: void TrackViewState();

24 [VB] Overrides Protected Sub TrackViewState()

25 [JScript] protected override function TrackViewState();

1  
2 *Description*

3 Marks the starting point to begin tracking and saving changes to the control  
4 as part of the control viewstate.

5 CalendarDay class (System.Web.UI.WebControls)

6 TrackViewState

7  
8  
9 *Description*

10 Represents a date in the **System.Web.UI.WebControls.Calendar** control.

11 A **System.Web.UI.WebControls.CalendarDay** object represents a date in  
12 the **System.Web.UI.WebControls.Calendar** control. You can use this class in  
13 the **System.Web.UI.WebControls.Calendar.DayRender** event handler to  
14 programmatically access the properties of a date as it is rendered on the  
15 **System.Web.UI.WebControls.Calendar** control. This allows you to determine  
16 the properties of the day (such as whether the date is selectable, selected, today's  
17 date, or a weekend date) and programmatically control the appearance or behavior  
18 of the day.

19 CalendarDay

20 *Example Syntax:*

21 TrackViewState

22  
23 [C#] public CalendarDay(DateTime date, bool isWeekend, bool isToday, bool  
24 isSelected, bool isOtherMonth, string dayNumberText);

25 [C++] public: CalendarDay(DateTime date, bool isWeekend, bool isToday, bool

```

1  isSelected, bool isOtherMonth, String* dayNumberText);
2  [VB] Public Sub New(ByVal date As DateTime, ByVal isWeekend As Boolean,
3  ByVal isToday As Boolean, ByVal isSelected As Boolean, ByVal isOtherMonth
4  As Boolean, ByVal dayNumberText As String)
5  [JScript] public function CalendarDay(date : DateTime, isWeekend : Boolean,
6  isToday : Boolean, isSelected : Boolean, isOtherMonth : Boolean,
7  dayNumberText : String);
8

```

### *Description*

Initializes a new instance of the **System.Web.UI.WebControls.CalendarDay** class.

The following table shows initial property values for an instance of **System.Web.UI.WebControls.CalendarDay**. A **System.DateTime** object that contains the date represented by an instance of this class. **true** to indicate that the date represented by an instance of this class is either a Saturday or a Sunday; otherwise, **false**. **true** to indicate that the date represented by an instance of this class is the current date; otherwise, **false**. **true** to indicate that the date represented by an instance of this class is selected on the **System.Web.UI.WebControls.Calendar** control; otherwise, **false**. **true** to indicate that the date represented by an instance of this class is in a month other than the displayed month on the **System.Web.UI.WebControls.Calendar** control; otherwise, **false**. The day number for the date represented by this class.

Date

TrackViewState

1  
2 [C#] public DateTime Date {get;}

3 [C++] public: \_\_property DateTime get\_Date();

4 [VB] Public ReadOnly Property Date As DateTime

5 [JScript] public function get Date() : DateTime;

6  
7 *Description*

8 Gets the date represented by an instance of this class. This property is read-  
9 only.

10 Use the **System.Web.UI.WebControls.CalendarDay.Date** property to  
11 programmatically determine the date represented by an instance of this class.

12 DayNumberText

13 TrackViewState

14  
15 [C#] public string DayNumberText {get;}

16 [C++] public: \_\_property String\* get\_DayNumberText();

17 [VB] Public ReadOnly Property DayNumberText As String

18 [JScript] public function get DayNumberText() : String;

19  
20 *Description*

21 Gets the string equivalent of the day number for the date represented by an  
22 instance of the **System.Web.UI.WebControls.CalendarDay** class. This property  
23 is read-only.

24 Use the **System.Web.UI.WebControls.CalendarDay.DayNumberText**  
25 property to determine the string equivalent of the day number for the date

represented by an instance of this class. This allows you to programmatically control the appearance or behavior of the day, based on this value.

IsOtherMonth

TrackViewState

[C#] public bool IsOtherMonth {get;}

[C++] public: \_\_property bool get\_IsOtherMonth();

[VB] Public ReadOnly Property IsOtherMonth As Boolean

[JScript] public function get IsOtherMonth() : Boolean;

### *Description*

Gets a value that indicates whether the date represented by an instance of this class is in a month other than the month displayed in the

**System.Web.UI.WebControls.Calendar** control. This property is read-only.

Use the **System.Web.UI.WebControls.CalendarDay.IsOtherMonth** property to programmatically determine whether the date represented by an instance of this class is in a month other than the month displayed in the

**System.Web.UI.WebControls.Calendar** control. This allows you to

programmatically control the appearance or behavior of the day based on this value.

IsSelectable

TrackViewState

[C#] public bool IsSelectable {get; set;}

[C++] public: \_\_property bool get\_IsSelectable();public: \_\_property void



1 set\_IsSelectable(bool);

2 [VB] Public Property IsSelectable As Boolean

3 [JScript] public function get IsSelectable() : Boolean;public function set

4 IsSelectable(Boolean);

5  
6 *Description*

7 Gets or sets a value that indicates whether the date represented by an  
8 instance of this class can be selected in the  
9 **System.Web.UI.WebControls.Calendar** control.

10 Use the **System.Web.UI.WebControls.CalendarDay.IsSelectable**  
11 property to specify or determine whether the date represented by an instance of  
12 this class can be selected in the **System.Web.UI.WebControls.Calendar** control.  
13 This allows you to programmatically control the appearance behavior of the day,  
14 based on this value.

15 IsSelected

16 TrackViewState

17  
18 [C#] public bool IsSelected {get;}

19 [C++] public: \_\_property bool get\_IsSelected();

20 [VB] Public ReadOnly Property IsSelected As Boolean

21 [JScript] public function get IsSelected() : Boolean;

22  
23 *Description*

24

25

Gets a value that indicates whether the date represented by an instance of this class is selected in the **System.Web.UI.WebControls.Calendar** control. This property is read-only.

Use the **System.Web.UI.WebControls.CalendarDay.IsSelected** property to programmatically determine whether the date represented by an instance of this class is selected in the **System.Web.UI.WebControls.Calendar** control. This allows you to programmatically control the appearance or behavior of the day, based on this value.

IsToday

TrackViewState

[C#] public bool IsToday {get;}

[C++] public: \_\_property bool get\_IsToday();

[VB] Public ReadOnly Property IsToday As Boolean

[JScript] public function get IsToday() : Boolean;

### *Description*

Gets a value that indicates whether the date represented by an instance of this class is the same date specified by the

**System.Web.UI.WebControls.Calendar.TodaysDate** property of the

**System.Web.UI.WebControls.Calendar** control. This property is read-only.

Use the **System.Web.UI.WebControls.CalendarDay.IsToday** property to programmatically determine whether the date represented by an instance of this class is the same date specified by the

**System.Web.UI.WebControls.Calendar.TodaysDate** property of the

**System.Web.UI.WebControls.Calendar** control. This allows you to programmatically control the appearance or behavior of the day, based on this value.

**IsWeekend**

**TrackViewState**

[C#] public bool IsWeekend {get;}

[C++] public: \_\_property bool get\_IsWeekend();

[VB] Public ReadOnly Property IsWeekend As Boolean

[JScript] public function get IsWeekend() : Boolean;

### *Description*

Gets a value that indicates whether the date represented by an instance of this class is either Saturday or Sunday. This property is read-only.

Use the **System.Web.UI.WebControls.CalendarDay.IsWeekend** property to programmatically determine whether the date represented by an instance of this class is either a Saturday or a Sunday. This allows you to programmatically control the appearance or behavior of the day, based on this value.

**CalendarSelectionMode** enumeration (System.Web.UI.WebControls)

**ToString**

### *Description*

Specifies the date selection mode of the  
**System.Web.UI.WebControls.Calendar** control.

The **System.Web.UI.WebControls.CalendarSelectionMode** enumeration  
represents the date selection modes of the  
**System.Web.UI.WebControls.Calendar** control.

**ToString**

[C#] public const CalendarSelectionMode Day;  
[C++] public: const CalendarSelectionMode Day;  
[VB] Public Const Day As CalendarSelectionMode  
[JScript] public var Day : CalendarSelectionMode;

#### *Description*

A single date can be selected on the  
**System.Web.UI.WebControls.Calendar** control.

**ToString**

[C#] public const CalendarSelectionMode DayWeek;  
[C++] public: const CalendarSelectionMode DayWeek;  
[VB] Public Const DayWeek As CalendarSelectionMode  
[JScript] public var DayWeek : CalendarSelectionMode;

#### *Description*

A single day or entire week can be selected on the  
**System.Web.UI.WebControls.Calendar** control.

1 ToString

2

3 [C#] public const CalendarSelectionMode DayWeekMonth;

4 [C++] public: const CalendarSelectionMode DayWeekMonth;

5 [VB] Public Const DayWeekMonth As CalendarSelectionMode

6 [JScript] public var DayWeekMonth : CalendarSelectionMode;

7

8 *Description*

9 A single date, week, or entire month can be selected on the

10 **System.Web.UI.WebControls.Calendar** control.

11 ToString

12

13 [C#] public const CalendarSelectionMode None;

14 [C++] public: const CalendarSelectionMode None;

15 [VB] Public Const None As CalendarSelectionMode

16 [JScript] public var None : CalendarSelectionMode;

17

18 *Description*

19 No dates can be selected on the **System.Web.UI.WebControls.Calendar**

20 control.

21 TableRow.CellControlCollection class (System.Web.UI.WebControls)

22 ToString

23

24

25 *Description*

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

Count

IsReadOnly

IsSynchronized

Item

Owner

SyncRoot

Add

[C#] public override void Add(Control child);

[C++] public: void Add(Control\* child);

[VB] Overrides Public Sub Add(ByVal child As Control)

[JScript] public override function Add(child : Control);

### *Description*

Adds the specified **System.Web.UI.Control** object to the collection. The new control is added to the end of the array.

AddAt

[C#] public override void AddAt(int index, Control child);

[C++] public: void AddAt(int index, Control\* child);

[VB] Overrides Public Sub AddAt(ByVal index As Integer, ByVal child As Control)

[JScript] public override function AddAt(index : int, child : Control);

1  
2 *Description*

3 Adds the specified **System.Web.UI.Control** object to the collection. The  
4 new control is added to the array at the specified index location. The location in  
5 the array to add the child control. The **Control** object to add to the collection.

6 **CheckBox** class (System.Web.UI.WebControls)

7 **ToString**  
8  
9

10 *Description*

11 Displays a check box that allows the user to select a **true** or **false** condition.

12 Use the **System.Web.UI.WebControls.CheckBox** control to allow the  
13 user to select a **true** or **false** state.

14 **CheckBox**

15 *Example Syntax:*

16 **ToString**  
17

18 [C#] public **CheckBox**();

19 [C++] public: **CheckBox**();

20 [VB] Public Sub **New**()

21 [JScript] public function **CheckBox**();  
22

23 *Description*

24 Initializes a new instance of the **System.Web.UI.WebControls.CheckBox**  
25 class.

1 Use this constructor to create and initialize a new instance of the  
2 **System.Web.UI.WebControls.CheckBox** class.

3 AccessKey

4 Attributes

5 AutoPostBack

6 ToString

7  
8  
9 *Description*

10 Gets or sets a value indicating whether the  
11 **System.Web.UI.WebControls.CheckBox** state automatically posts back to the  
12 server when clicked.

13 Use this property to specify whether the state of the  
14 **System.Web.UI.WebControls.CheckBox** control is posted back to the server  
15 when clicked.

16 BackColor

17 BorderColor

18 BorderStyle

19 BorderWidth

20 Checked

21 ToString

22  
23  
24 *Description*  
25



1 Gets or sets a value indicating whether the  
2 **System.Web.UI.WebControls.CheckBox** control is checked.  
3 Use this property to determine whether the  
4 **System.Web.UI.WebControls.CheckBox** control is checked. This property can  
5 also be used to programmatically set the state of the  
6 **System.Web.UI.WebControls.CheckBox** control.

7 ChildControlsCreated

8 ClientID

9 Context

10 Controls

11 ControlStyle .

12 ControlStyleCreated

13 CssClass

14 Enabled

15 EnableViewState

16 Events

17 Font

18 ForeColor

19 HasChildViewState

20 Height

21 ID

22 IsTrackingViewState

23 NamingContainer

24 Page

25 Parent

1 Site  
2 Style  
3 TabIndex  
4 TagKey  
5 TagName  
6 TemplateSourceDirectory  
7 Text  
8 ToString

9  
10  
11 *Description*

12 Gets or sets the text label associated with the  
13 **System.Web.UI.WebControls.CheckBox** .

14 Use this property to specify the text label associated with the  
15 **System.Web.UI.WebControls.CheckBox** control. This property can also be used  
16 to programmatically get the text label associated with the  
17 **System.Web.UI.WebControls.CheckBox** control.

18 TextAlign

19 ToString

20  
21 [C#] public virtual TextAlign TextAlign {get; set;}

22 [C++] public: \_\_property virtual TextAlign get\_TextAlign();public: \_\_property

23 virtual void set\_TextAlign(TextAlign);

24 [VB] Overridable Public Property TextAlign As TextAlign

25 [JScript] public function get TextAlign() : TextAlign;public function set

1 TextAlign(TextAlign);

2  
3 *Description*

4 Gets or sets the alignment of the text label associated with the  
5 **System.Web.UI.WebControls.CheckBox** control.

6 Use this property to specify the alignment of the text label associated with  
7 the **System.Web.UI.WebControls.CheckBox** control. You can specify whether  
8 the text label appears to the right or left of the check box. Use the  
9 **System.Web.UI.WebControls.CheckBox.Text** property to specify the label text.

10 ToolTip

11 UniqueID

12 ViewState

13 ViewStateIgnoresCase

14 Visible

15 Width

16 ToString

17  
18  
19 *Description*

20 Occurs when the **System.Web.UI.WebControls.CheckBox.Checked**  
21 property is changed.

22 The **System.Web.UI.WebControls.CheckBox.CheckedChanged** event is  
23 raised when the **System.Web.UI.WebControls.CheckBox.Checked** property is  
24 changed.

25 OnCheckedChanged

1  
2 [C#] protected virtual void OnCheckedChanged(EventArgs e);  
3 [C++] protected: virtual void OnCheckedChanged(EventArgs\* e);  
4 [VB] Overridable Protected Sub OnCheckedChanged(ByVal e As EventArgs)  
5 [JScript] protected function OnCheckedChanged(e : EventArgs);  
6

7 *Description*

8       Raises the **System.Web.UI.WebControls.CheckBox.CheckedChanged**  
9 event of the **System.Web.UI.WebControls.CheckBox** control. This allows you  
10 to handle the event directly.

11       The **System.Web.UI.WebControls.CheckBox.CheckedChanged** event is  
12 raised when the **System.Web.UI.WebControls.CheckBox.Checked** property is  
13 changed. A **System.EventArgs** that contains the event data.

14       **OnPreRender**

15  
16 [C#] protected override void OnPreRender(EventArgs e);  
17 [C++] protected: void OnPreRender(EventArgs\* e);  
18 [VB] Overrides Protected Sub OnPreRender(ByVal e As EventArgs)  
19 [JScript] protected override function OnPreRender(e : EventArgs);  
20

21 *Description*

22       Registers client script for generating postback prior to rendering on the  
23 client if **System.Web.UI.WebControls.CheckBox.AutoPostBack** is **true** .

24       **Render**

```

1
2 [C#] protected override void Render(HtmlTextWriter writer);
3 [C++] protected: void Render(HtmlTextWriter* writer);
4 [VB] Overrides Protected Sub Render(ByVal writer As HtmlTextWriter)
5 [JScript] protected override function Render(writer : HtmlTextWriter);
6

```

### *Description*

Displays the **System.Web.UI.WebControls.CheckBox** on the client. A **System.Web.UI.HtmlTextWriter** that contains the output stream to render on the client.

**IPostBackDataHandler.LoadPostData**

```

13 [C#] bool IPostBackDataHandler.LoadPostData(string postDataKey,
14 NameValueCollection postCollection);
15 [C++] bool IPostBackDataHandler::LoadPostData(String* postDataKey,
16 NameValueCollection* postCollection);
17 [VB] Function LoadPostData(ByVal postDataKey As String, ByVal
18 postCollection As NameValueCollection) As Boolean Implements
19 IPostBackDataHandler.LoadPostData
20 [JScript] function IPostBackDataHandler.LoadPostData(postDataKey : String,
21 postCollection : NameValueCollection) : Boolean;

```

**IPostBackDataHandler.RaisePostDataChangedEvent**

```

24 [C#] void IPostBackDataHandler.RaisePostDataChangedEvent();
25 [C++] void IPostBackDataHandler::RaisePostDataChangedEvent();

```

1 [VB] Sub RaisePostDataChangedEvent() Implements

2 IPostBackDataHandler.RaisePostDataChangedEvent

3 [JScript] function IPostBackDataHandler.RaisePostDataChangedEvent();

4     CheckBoxList class (System.Web.UI.WebControls)

5     TrackViewState

6  
7  
8 *Description*

9     Creates a multi selection check box group that can be dynamically created  
10 by binding the control to a data source.

11     The **System.Web.UI.WebControls.CheckBoxList** control provides a  
12 multi selection check box group that can be dynamically generated with data  
13 binding. It contains an **System.Web.UI.WebControls.ListControl.Items**  
14 collection with members that correspond to individual items in the list. To  
15 determine which items are checked, loop through and test the  
16 **System.Web.UI.WebControls.ListItem.Selected** property of each item in the  
17 list.

18     CheckBoxList

19     *Example Syntax:*

20     TrackViewState

21  
22 [C#] public CheckBoxList();

23 [C++] public: CheckBoxList();

24 [VB] Public Sub New()

25 [JScript] public function CheckBoxList();

1  
2 *Description*

3       Initializes a new instance of the  
4 **System.Web.UI.WebControls.CheckBoxList** class.

5       Use this constructor to create and initialize a new instance of the  
6 **System.Web.UI.WebControls.CheckBoxList** class.

7       AccessKey

8       Attributes

9       AutoPostBack

10      BackColor

11      BorderColor

12      BorderStyle

13      BorderWidth

14      CellPadding

15      TrackViewState

16  
17  
18 *Description*

19       Gets or sets the distance (in pixels) between the border and contents of the  
20 cell.

21       Use this property to control the spacing between the contents of a cell and  
22 the cell's border in the **System.Web.UI.WebControls.CheckBoxList** control.

23      CellSpacing

24      TrackViewState

```

1
2 [C#] public virtual int CellSpacing {get; set;}
3 [C++] public: __property virtual int get_CellSpacing();public: __property virtual
4 void set_CellSpacing(int);
5 [VB] Overridable Public Property CellSpacing As Integer
6 [JScript] public function get CellSpacing() : int;public function set
7 CellSpacing(int);
8

```

### 9 *Description*

10 Gets or sets the distance (in pixels) between cells.

11 Use this property to control the spacing between individual cells in the  
12 **System.Web.UI.WebControls.CheckBoxList** control. This property is applied  
13 both vertically and horizontally.

14 ChildControlsCreated

15 ClientID

16 Context

17 Controls

18 ControlStyle

19 ControlStyleCreated

20 CssClass

21 DataMember

22 DataSource

23 DataTextField

24 DataTextFormatString

25 DataValueField





## TrackViewState

```
[C#] public virtual RepeatDirection RepeatDirection {get; set;}
[C++] public: __property virtual RepeatDirection get_RepeatDirection();public:
__property virtual void set_RepeatDirection(RepeatDirection);
[VB] Overridable Public Property RepeatDirection As RepeatDirection
[JavaScript] public function get RepeatDirection() : RepeatDirection;public function
set RepeatDirection(RepeatDirection);
```

### *Description*

Gets or sets a value that indicates whether the control displays vertically or horizontally.

Use this property to specify the display direction of the **System.Web.UI.WebControls.CheckBoxList** control.

RepeatLayout

TrackViewState

```
[C#] public virtual RepeatLayout RepeatLayout {get; set;}
[C++] public: __property virtual RepeatLayout get_RepeatLayout();public:
__property virtual void set_RepeatLayout(RepeatLayout);
[VB] Overridable Public Property RepeatLayout As RepeatLayout
[JavaScript] public function get RepeatLayout() : RepeatLayout;public function set
RepeatLayout(RepeatLayout);
```

### *Description*

1 Gets or sets the layout of the check boxes.

2 Use this property to specify whether the items in the  
3 **System.Web.UI.WebControls.CheckBoxList** control are displayed in a table. If  
4 this property is set to **RepeatLayout.Table** , the items in the list are displayed in a  
5 table. If this property is set to **RepeatLayout.Flow** , the items in the list are  
6 displayed without a table structure.

7 SelectedIndex

8 SelectedItem

9 Site

10 Style

11 TabIndex

12 TagKey

13 TagName

14 TemplateSourceDirectory

15 TextAlign

16 TrackViewState

17  
18  
19 *Description*

20 Gets or sets the text alignment for the check boxes within the group.

21 Use this property to specify whether the text associated with the check  
22 boxes appears on the left or right of the check box. If this property is set to  
23 **TextAlign.Right** , the text is displayed to the right of the check box. If this  
24 property is set to **TextAlign.Left** , the text is displayed to the left of the check box.

25 ToolTip

1 UniqueID  
2 ViewState  
3 ViewStateIgnoresCase  
4 Visible  
5 Width  
6 CreateControlStyle  
7  
8 [C#] protected override Style CreateControlStyle();  
9 [C++] protected: Style\* CreateControlStyle();  
10 [VB] Overrides Protected Function CreateControlStyle() As Style  
11 [JScript] protected override function CreateControlStyle() : Style;

### 13 *Description*

14 Creates a new control style object.

15 *Return Value:* A **System.Web.UI.WebControls.Style** that contains the style  
16 properties of a control.

### 17 FindControl

18  
19 [C#] protected override Control FindControl(string id, int pathOffset);  
20 [C++] protected: Control\* FindControl(String\*.id, int pathOffset);  
21 [VB] Overrides Protected Function FindControl(ByVal id As String, ByVal  
22 pathOffset As Integer) As Control  
23 [JScript] protected override function FindControl(id : String, pathOffset : int) :  
24 Control;

1  
2 *Description*

3 Catches post data for each **System.Web.UI.WebControls.CheckBox** in  
4 the list.

5 **OnPreRender**

6  
7 [C#] protected override void OnPreRender(EventArgs e);

8 [C++] protected: void OnPreRender(EventArgs\* e);

9 [VB] Overrides Protected Sub OnPreRender(ByVal e As EventArgs)

10 [JScript] protected override function OnPreRender(e : EventArgs);

11  
12 *Description*

13 Configures the **System.Web.UI.WebControls.CheckBoxList** prior to  
14 rendering on the client.

15 **Render**

16  
17 [C#] protected override void Render(HtmlTextWriter writer);

18 [C++] protected: void Render(HtmlTextWriter\* writer);

19 [VB] Overrides Protected Sub Render(ByVal writer As HtmlTextWriter)

20 [JScript] protected override function Render(writer : HtmlTextWriter);

21  
22 *Description*

23 Displays the **System.Web.UI.WebControls.CheckBoxList** on the client.

24 A **System.Web.UI.HtmlTextWriter** that contains the output stream for rendering  
25 on the client.

```

1      IPostBackDataHandler.LoadPostData
2
3      [C#] bool IPostBackDataHandler.LoadPostData(string postDataKey,
4      NameValueCollection postCollection);
5      [C++] bool IPostBackDataHandler::LoadPostData(String* postDataKey,
6      NameValueCollection* postCollection);
7      [VB] Function LoadPostData(ByVal postDataKey As String, ByVal
8      postCollection As NameValueCollection) As Boolean Implements
9      IPostBackDataHandler.LoadPostData
10     [JScript] function IPostBackDataHandler.LoadPostData(postDataKey : String,
11     postCollection : NameValueCollection) : Boolean;
12
13     IPostBackDataHandler.RaisePostDataChangedEvent
14
15     [C#] void IPostBackDataHandler.RaisePostDataChangedEvent();
16     [C++] void IPostBackDataHandler::RaisePostDataChangedEvent();
17     [VB] Sub RaisePostDataChangedEvent() Implements
18     IPostBackDataHandler.RaisePostDataChangedEvent
19     [JScript] function IPostBackDataHandler.RaisePostDataChangedEvent();
20
21     IRepeatInfoUser.GetItemStyle
22
23     [C#] Style IRepeatInfoUser.GetItemStyle(ListItemType itemType, int
24     repeatIndex);
25     [C++] Style* IRepeatInfoUser::GetItemStyle(ListItemType itemType, int
26     repeatIndex);
27     [VB] Function GetItemStyle(ByVal itemType As ListItemType, ByVal

```

```

1 repeatIndex As Integer) As Style Implements IRepeatInfoUser.GetItemStyle
2 [JScript] function IRepeatInfoUser.GetItemStyle(itemType : ListItemType,
3 repeatIndex : int) : Style;
4         IRepeatInfoUser.RenderItem
5
6 [C#] void IRepeatInfoUser.RenderItem(ListItemType itemType, int repeatIndex,
7 RepeatInfo repeatInfo, HtmlTextWriter writer);
8 [C++] void IRepeatInfoUser::RenderItem(ListItemType itemType, int
9 repeatIndex, RepeatInfo* repeatInfo, HtmlTextWriter* writer);
10 [VB] Sub RenderItem(ByVal itemType As ListItemType, ByVal repeatIndex As
11 Integer, ByVal repeatInfo As RepeatInfo, ByVal writer As HtmlTextWriter)
12 Implements IRepeatInfoUser.RenderItem
13 [JScript] function IRepeatInfoUser.RenderItem(itemType : ListItemType,
14 repeatIndex : int, repeatInfo : RepeatInfo, writer : HtmlTextWriter);
15         CommandEventArgs class (System.Web.UI.WebControls)
16         ViewState
17
18
19

```

### *Description*

Provides data for the **Command** event.

The **Command** event is raised when a

**System.Web.UI.WebControls.Button** or

**System.Web.UI.WebControls.ImageButton** control is clicked.

CommandEventArgs

*Example Syntax:*

## TrackViewState

```
[C#] public CommandEventArgs(CommandEventArgs e);  
[C++] public: CommandEventArgs(CommandEventArgs* e);  
[VB] Public Sub New(ByVal e As CommandEventArgs)  
[JScript] public function CommandEventArgs(e : CommandEventArgs);
```

Initializes a new instance of the

**System.Web.UI.WebControls.CommandEventArgs** class.

### *Description*

Initializes a new instance of the

**System.Web.UI.WebControls.CommandEventArgs** class with another

**System.Web.UI.WebControls.CommandEventArgs** object.

Use this constructor to create and initialize a new instance of the

**System.Web.UI.WebControls.CommandEventArgs** class using the specified

**System.Web.UI.WebControls.CommandEventArgs** object. A

**System.Web.UI.WebControls.CommandEventArgs** that contains the event data.

**CommandEventArgs**

### *Example Syntax:*

**TrackViewState**

```
[C#] public CommandEventArgs(string commandName, object argument);  
[C++] public: CommandEventArgs(String* commandName, Object* argument);  
[VB] Public Sub New(ByVal commandName As String, ByVal argument As
```



Object)

[JScript] public function CommandEventArgs(commandName : String, argument : Object);

### *Description*

Initializes a new instance of the **System.Web.UI.WebControls.CommandEventArgs** class with the specified command name and argument.

Use this constructor to create and initialize a new instance of the **System.Web.UI.WebControls.CommandEventArgs** class using the specified command name and argument. The name of the command. A **System.Object** that contains the arguments for the command.

CommandArgument

TrackViewState

[C#] public object CommandArgument {get;}

[C++] public: \_\_property Object\* get\_CommandArgument();

[VB] Public ReadOnly Property CommandArgument As Object

[JScript] public function get CommandArgument() : Object;

### *Description*

Gets the argument for the command.

The

**System.Web.UI.WebControls.CommandEventArgs.CommandArgument** can contain any string set by the programmer. The

**System.Web.UI.WebControls.CommandEventArgs.CommandArgument**

property complements the

**System.Web.UI.WebControls.CommandEventArgs.CommandName** property

by allowing you to provide any additional information for the command. For

example, you can set the

**System.Web.UI.WebControls.CommandEventArgs.CommandName** property

to **Sort** and set the

**System.Web.UI.WebControls.CommandEventArgs.CommandArgument**

property to **Ascending** to specify a command to sort in ascending order.

CommandName

TrackViewState

[C#] public string CommandName {get;}

[C++] public: \_\_property String\* get\_CommandName();

[VB] Public ReadOnly Property CommandName As String

[JScript] public function get CommandName() : String;

### *Description*

Gets the name of the command.

Use the

**System.Web.UI.WebControls.CommandEventArgs.CommandName** property

to determine the command to perform. The

**System.Web.UI.WebControls.CommandEventArgs.CommandName** property

can contain any string set by the programmer. The programmer can then identify

the command name in code and perform the appropriate tasks.

1 CommandEventHandler delegate (System.Web.UI.WebControls)

2 ToString

3  
4  
5 *Description*

6 Represents the method that will handle the **Command** event. The source of  
7 the event. A **System.Web.UI.WebControls.CommandEventArgs** that contains  
8 the event data.

9 The **Command** event is raised when a  
10 **System.Web.UI.WebControls.Button** or  
11 **System.Web.UI.WebControls.ImageButton** control is clicked.

12 CompareValidator class (System.Web.UI.WebControls)

13 ToString

14  
15  
16 *Description*

17 Compares the value entered by the user into an input control with the value  
18 entered into another input control or a constant value.

19 Use the **System.Web.UI.WebControls.CompareValidator** control to  
20 compare the value entered by the user into an input control, such as a  
21 **System.Web.UI.WebControls.TextBox** control, with the value entered into  
22 another input control or a constant value. You can also use the  
23 **System.Web.UI.WebControls.CompareValidator** control to indicate whether  
24 the value entered into an input control can be converted to the data type specified  
25 by the **System.Web.UI.WebControls.BaseCompareValidator.Type** property.

1	CompareValidator
2	<i>Example Syntax:</i>
3	ToString
4	
5	[C#] public CompareValidator();
6	[C++] public: CompareValidator();
7	[VB] Public Sub New()
8	[JScript] public function CompareValidator();
9	AccessKey
10	Attributes
11	BackColor
12	BorderColor
13	BorderStyle
14	BorderWidth
15	ChildControlsCreated
16	ClientID
17	Context
18	Controls
19	ControlStyle
20	ControlStyleCreated
21	ControlToCompare
22	ToString
23	
24	
25	<i>Description</i>

1 Gets or sets the input control to compare with the input control being  
2 validated.

3 Use the  
4 **System.Web.UI.WebControls.CompareValidator.ControlToCompare**  
5 property to specify an input control, such as a  
6 **System.Web.UI.WebControls.TextBox** , to compare with the input control being  
7 validated. If the input control specified by this property is not a control on the  
8 page, an exception is thrown.

9 ControlToValidate

10 CssClass

11 Display

12 EnableClientScript

13 Enabled

14 EnableViewState

15 ErrorMessage

16 Events

17 Font

18 ForeColor

19 HasChildViewState

20 Height

21 ID

22 IsTrackingViewState

23 IsValid

24 NamingContainer

25 Operator

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

ToString

*Description*

Gets or sets the comparison operation to perform.

Use the **System.Web.UI.WebControls.CompareValidator.Operator** property to specify the comparison operation to perform. The following table lists the comparison operations that are possible.

- Page
- Parent
- PropertiesValid
- RenderUplevel
- Site
- Style
- TabIndex
- TagKey
- TagName
- TemplateSourceDirectory
- Text
- ToolTip
- Type
- UniqueID
- ValueToCompare
- ToString

1  
2  
3 *Description*

4 Gets or sets a constant value to compare with the value entered by the user  
5 into the input control being validated.

6 Use the  
7 **System.Web.UI.WebControls.CompareValidator.ValueToCompare** property  
8 to specify a constant value to compare with the value entered by the user into the  
9 input control being validated. If the constant value specified by this property fails  
10 to convert to the data type specified by the  
11 **System.Web.UI.WebControls.BaseCompareValidator.Type** property, a  
12 exception is thrown.

13 ViewState

14 ViewStateIgnoresCase

15 Visible

16 Width

17 AddAttributesToRender

18  
19 [C#] protected override void AddAttributesToRender(HtmlTextWriter writer);

20 [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);

21 [VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As

22 HtmlTextWriter)

23 [JScript] protected override function AddAttributesToRender(writer :

24 HtmlTextWriter);

1  
2 *Description*

3 Adds the attributes of this control to the output stream for rendering on the  
4 client. A **System.Web.UI.HtmlTextWriter** that contains the output stream for  
5 rendering on the client.

6 **ControlPropertiesValid**

7  
8 [C#] protected override bool ControlPropertiesValid();

9 [C++] protected: bool ControlPropertiesValid();

10 [VB] Overrides Protected Function ControlPropertiesValid() As Boolean

11 [JScript] protected override function ControlPropertiesValid() : Boolean;

12  
13 *Description*

14 Checks the properties of a the control for valid values.

15 *Return Value:* **true** if the control properties are valid; otherwise **false** .

16 **EvaluateIsValid**

17  
18 [C#] protected override bool EvaluateIsValid();

19 [C++] protected: bool EvaluateIsValid();

20 [VB] Overrides Protected Function EvaluateIsValid() As Boolean

21 [JScript] protected override function EvaluateIsValid() : Boolean;

22  
23 *Description*

24 **EvaluateIsValid** method **EvaluateIsValid** method

25 **CustomValidator** class (System.Web.UI.WebControls)



## Validate

### *Description*

Performs user-defined validation on an input control.

Use the **System.Web.UI.WebControls.CustomValidator** control to provide a user-defined validation function for an input control. The **System.Web.UI.WebControls.CustomValidator** control is separated from the input control it validates, which allows you to control where the validation message is displayed.

CustomValidator

### *Example Syntax:*

Validate

```
[C#] public CustomValidator();
```

```
[C++] public: CustomValidator();
```

```
[VB] Public Sub New()
```

```
[JScript] public function CustomValidator();
```

AccessKey

Attributes

BackColor

BorderColor

BorderStyle

BorderWidth

ChildControlsCreated

1 ClientID  
2 ClientValidationFunction  
3 Validate  
4  
5

6 *Description*

7 Gets or sets the name of the custom client-side script function used for  
8 validation.

9 Set this property to the name of the function that performs the client-side  
10 validation.

11 Context

12 Controls

13 ControlStyle

14 ControlStyleCreated

15 ControlToValidate

16 CssClass

17 Display

18 EnableClientScript

19 Enabled

20 EnableViewState

21 ErrorMessage

22 Events

23 Font

24 ForeColor

25 HasChildViewState

1	Height
2	ID
3	IsTrackingViewState
4	IsValid
5	NamingContainer
6	Page
7	Parent
8	PropertiesValid
9	RenderUplevel
10	Site
11	Style
12	TabIndex
13	TagKey
14	TagName
15	TemplateSourceDirectory
16	Text
17	ToolTip
18	UniqueID
19	ViewState
20	ViewStateIgnoresCase
21	Visible
22	Width
23	Validate
24	
25	

1  
2  
3 *Description*

4 Occurs when validation is performed on the server.

5 The **System.Web.UI.WebControls.CustomValidator.ServerValidate**  
6 event is raised when validation is performed on the server. This event is used to  
7 provide a custom validation routine for an input control, such as a  
8 **System.Web.UI.WebControls.TextBox** control.

9 AddAttributesToRender

10  
11 [C#] protected override void AddAttributesToRender(HtmlTextWriter writer);

12 [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);

13 [VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As  
14 HtmlTextWriter)

15 [JScript] protected override function AddAttributesToRender(writer :  
16 HtmlTextWriter);

17  
18 *Description*

19 Adds the properties of the  
20 **System.Web.UI.WebControls.CustomValidator** control to the output stream for  
21 rendering on the client. A **System.Web.UI.HtmlTextWriter** that contains the  
22 output stream for rendering on the client.

23 ControlPropertiesValid

24  
25 [C#] protected override bool ControlPropertiesValid();

1 [C++] protected: bool ControlPropertiesValid();

2 [VB] Overrides Protected Function ControlPropertiesValid() As Boolean

3 [JScript] protected override function ControlPropertiesValid() : Boolean;

4  
5 *Description*

6 Checks the properties of the control for valid values.

7 *Return Value:* **true** if the control properties are valid; otherwise **false** .

8 EvaluateIsValid

9  
10 [C#] protected override bool EvaluateIsValid();

11 [C++] protected: bool EvaluateIsValid();

12 [VB] Overrides Protected Function EvaluateIsValid() As Boolean

13 [JScript] protected override function EvaluateIsValid() : Boolean;

14  
15 *Description*

16 EvaluateIsValid method EvaluateIsValid method

17 OnServerValidate

18  
19 [C#] protected virtual bool OnServerValidate(string value);

20 [C++] protected: virtual bool OnServerValidate(String\* value);

21 [VB] Overridable Protected Function OnServerValidate(ByVal value As String)

22 As Boolean

23 [JScript] protected function OnServerValidate(value : String) : Boolean;

24  
25 *Description*

1       Raises the  
2       **System.Web.UI.WebControls.CustomValidator.ServerValidate** event for the  
3       **System.Web.UI.WebControls.CustomValidator** control.

4       The **System.Web.UI.WebControls.CustomValidator.ServerValidate**  
5       event is raised when validation is performed on the server. The value to validate.

6       DataGrid class (System.Web.UI.WebControls)

7       Validate

8  
9  
10      *Description*

11       A data bound list control that displays the items from data source in a table.  
12       The **System.Web.UI.WebControls.DataGrid** control allows you to select, sort,  
13       and edit these items.

14       Use the **System.Web.UI.WebControls.DataGrid** control to display the  
15       fields of a data source as columns in a table. Each row in the  
16       **System.Web.UI.WebControls.DataGrid** control represents a record in the data  
17       source. The **System.Web.UI.WebControls.DataGrid** control supports selection,  
18       editing, deleting, paging, and sorting.

19       Validate

20  
21      [C#] public const string CancelCommandName;

22      [C++] public: const String\* CancelCommandName;

23      [VB] Public Const CancelCommandName As String

24      [JavaScript] public var CancelCommandName : String;

1  
2 *Description*

3 Represents the **Cancel** command name. This field is read-only.

4 Use the

5 **System.Web.UI.WebControls.DataGrid.CancelCommandName** field to  
6 represent the **Cancel** command name.

7 Validate

8  
9 [C#] public const string DeleteCommandName;

10 [C++] public: const String\* DeleteCommandName;

11 [VB] Public Const DeleteCommandName As String

12 [JScript] public var DeleteCommandName : String;

13  
14 *Description*

15 Represents the **Delete** command name. This field is read-only.

16 Use the **System.Web.UI.WebControls.DataGrid.DeleteCommandName**  
17 field to represent the **Delete** command name.

18 Validate

19  
20 [C#] public const string EditCommandName;

21 [C++] public: const String\* EditCommandName;

22 [VB] Public Const EditCommandName As String

23 [JScript] public var EditCommandName : String;

24  
25 *Description*

Represents the **Edit** command name. This field is read-only.

Use the **System.Web.UI.WebControls.DataGrid.EditCommandName** field to represent the **Edit** command name.

Validate

[C#] public const string NextPageCommandArgument;

[C++] public: const String\* NextPageCommandArgument;

[VB] Public Const NextPageCommandArgument As String

[JScript] public var NextPageCommandArgument : String;

#### *Description*

Represents the **Next** command argument. This field is read-only.

Use the

**System.Web.UI.WebControls.DataGrid.NextPageCommandArgument** field to represent the **Next** command argument.

Validate

[C#] public const string PageCommandName;

[C++] public: const String\* PageCommandName;

[VB] Public Const PageCommandName As String

[JScript] public var PageCommandName : String;

#### *Description*

Represents the **Page** command name. This field is read-only.



1        Use the **System.Web.UI.WebControls.DataGrid.PageCommandName**  
2 field to represent the **Page** command name.

3        Validate

4  
5        [C#] public const string PrevPageCommandArgument;

6        [C++] public: const String\* PrevPageCommandArgument;

7        [VB] Public Const PrevPageCommandArgument As String

8        [JScript] public var PrevPageCommandArgument : String;

9  
10        *Description*

11        Represents the **Prev** command argument. This field is read-only.

12        Use the

13        **System.Web.UI.WebControls.DataGrid.PrevPageCommandArgument** field

14 to represent the **Prev** command argument.

15        Validate

16  
17        [C#] public const string SelectCommandName;

18        [C++] public: const String\* SelectCommandName;

19        [VB] Public Const SelectCommandName As String

20        [JScript] public var SelectCommandName : String;

21  
22        *Description*

23        Represents the **Select** command name. This field is read-only.

24        Use the **System.Web.UI.WebControls.DataGrid.SelectCommandName**

25 field to represent the **Select** command name.

## Validate

```
[C#] public const string SortCommandName;  
[C++] public: const String* SortCommandName;  
[VB] Public Const SortCommandName As String  
[JScript] public var SortCommandName : String;
```

### *Description*

Represents the **Sort** command name. This field is read-only.

Use the **System.Web.UI.WebControls.DataGrid.SortCommandName** field to represent the **Sort** command name.

## Validate

```
[C#] public const string UpdateCommandName;  
[C++] public: const String* UpdateCommandName;  
[VB] Public Const UpdateCommandName As String  
[JScript] public var UpdateCommandName : String;
```

### *Description*

Represents the **Update** command name. This field is read-only.

Use the **System.Web.UI.WebControls.DataGrid.UpdateCommandName** field to represent the **Update** command name.

## DataGrid

### *Example Syntax:*

1       Validate

2  
3   [C#] public DataGrid();

4   [C++] public: DataGrid();

5   [VB] Public Sub New()

6   [JavaScript] public function DataGrid();

7  
8   *Description*

9       Initializes a new instance of the **System.Web.UI.WebControls.DataGrid**  
10   class.

11       Use this constructor to create and initialize a new instance of the  
12   **System.Web.UI.WebControls.DataGrid** class.

13       AccessKey

14       AllowCustomPaging

15       Validate

16  
17  
18   *Description*

19       Gets or sets a value that indicates whether custom paging is enabled.

20       Paging allows you to display the contents of the  
21   **System.Web.UI.WebControls.DataGrid** control in page segments. The number  
22   of items on a page is determined by the  
23   **System.Web.UI.WebControls.DataGrid.PageSize** property. If no value is  
24   specified for the **System.Web.UI.WebControls.DataGrid.PageSize** property, the  
25   **System.Web.UI.WebControls.DataGrid** will display 10 items on a page.

AllowPaging

Validate

[C#] public virtual bool AllowPaging {get; set;}

[C++] public: \_\_property virtual bool get\_AllowPaging();public: \_\_property

virtual void set\_AllowPaging(bool);

[VB] Overridable Public Property AllowPaging As Boolean

[JScript] public function get AllowPaging() : Boolean;public function set

AllowPaging(Boolean);

### *Description*

Gets or sets a value that indicates whether paging is enabled.

Paging allows you to display the contents of the

**System.Web.UI.WebControls.DataGrid** control in page segments. The number of items on a page is determined by the

**System.Web.UI.WebControls.DataGrid.PageSize** property. If no value is specified for the **System.Web.UI.WebControls.DataGrid.PageSize** property, the **System.Web.UI.WebControls.DataGrid** control will display 10 items on a page.

AllowSorting

Validate

[C#] public virtual bool AllowSorting {get; set;}

[C++] public: \_\_property virtual bool get\_AllowSorting();public: \_\_property

virtual void set\_AllowSorting(bool);

[VB] Overridable Public Property AllowSorting As Boolean

1 [JScript] public function get AllowSorting() : Boolean;public function set  
2 AllowSorting(Boolean);

3  
4 *Description*

5 Gets or sets a value that indicates whether sorting is enabled.

6 When sorting is enabled, **System.Web.UI.WebControls.LinkButton**  
7 controls are rendered in the heading section of each column where the  
8 **System.Web.UI.WebControls.DataGridColumn.SortExpression** property is  
9 set. These **System.Web.UI.WebControls.LinkButton** controls allow you to sort  
10 the **System.Web.UI.WebControls.DataGrid** control by the selected column. The  
11 only exception is when you use a  
12 **System.Web.UI.WebControls.TemplateColumn** column type with the  
13 **System.Web.UI.WebControls.TemplateColumn.HeaderTemplate** property set.  
14 In this case, you must provide a **System.Web.UI.WebControls.Button** control in  
15 the **System.Web.UI.WebControls.TemplateColumn.HeaderTemplate** of the  
16 column.

17 AlternatingItemStyle

18 Validate

19  
20 [C#] public virtual TableItemStyle AlternatingItemStyle {get;}

21 [C++] public: \_\_property virtual TableItemStyle\* get\_AlternatingItemStyle();

22 [VB] Overridable Public ReadOnly Property AlternatingItemStyle As  
23 TableItemStyle

24 [JScript] public function get AlternatingItemStyle() : TableItemStyle;

1  
2 *Description*

3 Gets the style properties for alternating items in the  
4 **System.Web.UI.WebControls.DataGrid** control.

5 Use the **System.Web.UI.WebControls.DataGrid.AlternatingItemStyle**  
6 property to provide a custom style for the alternating items in the  
7 **System.Web.UI.WebControls.DataGrid** control. Common style attributes that  
8 can be adjusted include forecolor, backcolor, font, and content alignment within  
9 the cell. Providing a different style enhances the appearance of the  
10 **System.Web.UI.WebControls.DataGrid** control.

11 Attributes

12 AutoGenerateColumns

13 Validate

14  
15  
16 *Description*

17 Gets or sets a value that indicates whether  
18 **System.Web.UI.WebControls.BoundColumn** objects are automatically created  
19 and displayed in the **System.Web.UI.WebControls.DataGrid** control for each  
20 field in the data source.

21 Use this property to automatically create a  
22 **System.Web.UI.WebControls.BoundColumn** object for each field in the data  
23 source. Each field is then rendered as a column in the  
24 **System.Web.UI.WebControls.DataGrid** control in the order that the fields  
25 appear in the data source.

1 BackColor  
2 BackImageUrl  
3 Validate  
4  
5

6 *Description*

7 Gets or sets the URL of an image to display in the background of the  
8 **System.Web.UI.WebControls.DataGrid** control.

9 Use the **System.Web.UI.WebControls.DataGrid.BackImageUrl** property  
10 to specify an image to display in the background of the  
11 **System.Web.UI.WebControls.DataGrid** control.

12 BorderColor  
13 BorderStyle  
14 BorderWidth  
15 CellPadding  
16 CellSpacing  
17 ChildControlsCreated  
18 ClientID  
19 Columns  
20 Validate  
21  
22

23 *Description*

24 Gets a collection of objects that represent the columns of the  
25 **System.Web.UI.WebControls.DataGrid** control.

1        Use this property to programmatically control the collection of columns in  
2 the **System.Web.UI.WebControls.DataGrid** control. The  
3 **System.Web.UI.WebControls.DataGrid.Columns** collection contains explicitly  
4 declared columns that get rendered in the  
5 **System.Web.UI.WebControls.DataGrid** control.

6        Context

7        Controls

8        ControlStyle

9        ControlStyleCreated

10       CssClass

11       CurrentPageIndex

12       Validate

13  
14  
15 *Description*

16       Gets or sets the index of the currently displayed page.

17       Use this property to determine the currently displayed page in the  
18 **System.Web.UI.WebControls.DataGrid** control when paging is enabled. This  
19 property is also used to programmatically control which page is displayed.

20       DataKeyField

21       DataKeys

22       DataKeysArray

23       DataMember

24       DataSource

25       EditItemIndex



## Validate

### *Description*

Gets or sets the index of an item in the **System.Web.UI.WebControls.DataGrid** control to edit.

When the **System.Web.UI.WebControls.DataGrid** control contains an **System.Web.UI.WebControls.EditCommandColumn** object, use this property to determine the index of the item selected in the **System.Web.UI.WebControls.DataGrid** control to edit.

### EditItemStyle

## Validate

[C#] public virtual TableItemStyle EditItemStyle {get;}

[C++] public: \_\_property virtual TableItemStyle\* get\_EditItemStyle();

[VB] Overridable Public ReadOnly Property EditItemStyle As TableItemStyle

[JScript] public function get EditItemStyle() : TableItemStyle;

### *Description*

Gets the style properties of the item selected for editing in the **System.Web.UI.WebControls.DataGrid** control.

Use the **System.Web.UI.WebControls.DataGrid.EditItemStyle** property to provide a custom style for the item selected for editing in the **System.Web.UI.WebControls.DataGrid** control. Common style attributes that can be adjusted include forecolor, backcolor, font, and content alignment within

1 the cell. Providing a different style enhances the appearance of the

2 **System.Web.UI.WebControls.DataGrid** control.

3 Enabled

4 EnableViewState

5 Events

6 Font

7 FooterStyle

8 Validate

9  
10  
11 *Description*

12 Gets the style properties of the footer section in the

13 **System.Web.UI.WebControls.DataGrid** control.

14 Use this property to provide a custom style for the footer of the  
15 **System.Web.UI.WebControls.DataGrid** control. Common style attributes that  
16 can be adjusted include forecolor, bgcolor, font, and content alignment within  
17 the cell. Providing a different style enhances the appearance of the

18 **System.Web.UI.WebControls.DataGrid** control.

19 ForeColor

20 GridLines

21 HasChildViewState

22 HeaderStyle

23 Validate

1  
2  
3 *Description*

4 Gets the style properties of the heading section in the  
5 **System.Web.UI.WebControls.DataGrid** control.

6 Use this property to provide a custom style for the heading section of the  
7 **System.Web.UI.WebControls.DataGrid** control. Common style attributes that  
8 can be adjusted include forecolor, backcolor, font, and content alignment within  
9 the cell. Providing a different style enhances the appearance of the  
10 **System.Web.UI.WebControls.DataGrid** control.

11 Height

12 HorizontalAlign

13 ID

14 IsTrackingViewState

15 Items

16 Validate

17  
18  
19 *Description*

20 Gets a collection of **System.Web.UI.WebControls.DataGridItem** objects  
21 that represent the individual items in the  
22 **System.Web.UI.WebControls.DataGrid** control.

23 Use the **System.Web.UI.WebControls.DataGrid.Items** collection to  
24 programmatically control the items in the  
25 **System.Web.UI.WebControls.DataGrid** control. The

**System.Web.UI.WebControls.DataGrid.Items** collection does not provide any methods to add or remove items to the collection. However, you can control the contents of an item by providing a handler for the **System.Web.UI.WebControls.DataGrid.ItemCreated** event.

ItemStyle

Validate

[C#] public virtual TableItemStyle ItemStyle {get;}

[C++] public: \_\_property virtual TableItemStyle\* get\_ItemStyle();

[VB] Overridable Public ReadOnly Property ItemStyle As TableItemStyle

[JScript] public function get ItemStyle() : TableItemStyle;

### *Description*

Gets the style properties of the items in the **System.Web.UI.WebControls.DataGrid** control.

Use this property to provide a custom style for the items of the **System.Web.UI.WebControls.DataGrid** control. Common style attributes that can be adjusted include forecolor, backcolor, font, and content alignment within the cell. Providing a different style enhances the appearance of the **System.Web.UI.WebControls.DataGrid** control.

NamingContainer

Page

PageCount

Validate

1  
2  
3 *Description*

4 Gets the total number of pages required to display the items in the  
5 **System.Web.UI.WebControls.DataGrid** control.

6 Use this property to programmatically determine the number of pages  
7 required to display the items in the **System.Web.UI.WebControls.DataGrid**  
8 control. This property is only used when the  
9 **System.Web.UI.WebControls.DataGrid.AllowPaging** property is set to **true** .

10 PagerStyle

11 Validate

12  
13 [C#] public virtual DataGridPagerStyle PagerStyle {get;}

14 [C++] public: \_\_property virtual DataGridPagerStyle\* get\_PagerStyle();

15 [VB] Overridable Public ReadOnly Property PagerStyle As DataGridPagerStyle

16 [JScript] public function get PagerStyle() : DataGridPagerStyle;

17  
18 *Description*

19 Gets the style properties of the paging section of the  
20 **System.Web.UI.WebControls.DataGrid** control.

21 Use this property to provide a custom style for the paging section of the  
22 **System.Web.UI.WebControls.DataGrid** control. Common style attributes that  
23 can be adjusted include forecolor, backcolor, font, and content alignment within  
24 the cell. Providing a different style enhances the appearance of the  
25 **System.Web.UI.WebControls.DataGrid** control.

1        **PageSize**

2        **Validate**

3  
4        [C#] public virtual int PageSize {get; set;}

5        [C++] public: \_\_property virtual int get\_PageSize();public: \_\_property virtual void  
6        set\_PageSize(int);

7        [VB] Overridable Public Property PageSize As Integer

8        [JScript] public function get PageSize() : int;public function set PageSize(int);

9  
10        *Description*

11            Gets or sets the number of items to display on a single page of the  
12        **System.Web.UI.WebControls.DataGrid** control.

13            Use this property to specify the number of items to display on a single page  
14        of the **System.Web.UI.WebControls.DataGrid** control. The  
15        **System.Web.UI.WebControls.DataGrid.AllowPaging** property must be set to  
16        **true** for this property have any effect.

17        **Parent**

18        **SelectedIndex**

19        **Validate**

20  
21  
22        *Description*

23            Gets or sets the index of the selected item in the  
24        **System.Web.UI.WebControls.DataGrid** control.

Use the **System.Web.UI.WebControls.DataGrid.SelectedIndex** property to determine the index of the item selected by the user in the **System.Web.UI.WebControls.DataGrid** control. You can also use this property to programmatically specify which item is selected in the **System.Web.UI.WebControls.DataGrid** control.

SelectedItem

Validate

[C#] public virtual DataGridItem SelectedItem {get;}

[C++] public: \_\_property virtual DataGridItem\* get\_SelectedItem();

[VB] Overridable Public ReadOnly Property SelectedItem As DataGridItem

[JScript] public function get SelectedItem() : DataGridItem;

#### *Description*

Gets a **System.Web.UI.WebControls.DataGridItem** object that represents the selected item in the **System.Web.UI.WebControls.DataGrid** control.

Use the **System.Web.UI.WebControls.DataGrid.SelectedItem** property to get a **System.Web.UI.WebControls.DataGridItem** object that represents the selected item in the **System.Web.UI.WebControls.DataGrid** control. This object can then be used to access the properties of the selected item.

SelectedItemStyle

Validate

[C#] public virtual TableItemStyle SelectedItemStyle {get;}

```

1 [C++] public: __property virtual TableItemStyle* get_SelectedItemStyle();
2 [VB] Overridable Public ReadOnly Property SelectedItemStyle As TableItemStyle
3 [JScript] public function get SelectedItemStyle() : TableItemStyle;

```

#### *Description*

Gets the style properties of the currently selected item in the **System.Web.UI.WebControls.DataGrid** control.

Use the **System.Web.UI.WebControls.DataGrid.SelectedItemStyle** property to provide a custom style for the selected item in the **System.Web.UI.WebControls.DataGrid** control. Common style attributes that can be adjusted include forecolor, backcolor, font, and content alignment within the cell. Providing a different style enhances the appearance of the **System.Web.UI.WebControls.DataGrid** control.

ShowFooter

Validate

```

17 [C#] public virtual bool ShowFooter {get; set;}
18 [C++] public: __property virtual bool get_ShowFooter();public: __property virtual
19 void set_ShowFooter(bool);
20 [VB] Overridable Public Property ShowFooter As Boolean
21 [JScript] public function get ShowFooter() : Boolean;public function set
22 ShowFooter(Boolean);

```

#### *Description*



1 Gets or sets a value that indicates whether the footer is displayed in the  
2 **System.Web.UI.WebControls.DataGrid** control.

3 Set this property to **true** to display the footer in the  
4 **System.Web.UI.WebControls.DataGrid** control. The appearance of the footer is  
5 controlled by using the **System.Web.UI.WebControls.DataGrid.FooterStyle**  
6 property.

7 ShowHeader

8 Validate

9  
10 [C#] public virtual bool ShowHeader {get; set;}

11 [C++] public: \_\_property virtual bool get\_ShowHeader();public: \_\_property  
12 virtual void set\_ShowHeader(bool);

13 [VB] Overridable Public Property ShowHeader As Boolean

14 [JScript] public function get ShowHeader() : Boolean;public function set  
15 ShowHeader(Boolean);

16  
17 *Description*

18 Gets or sets a value that indicates whether the header is displayed in the  
19 **System.Web.UI.WebControls.DataGrid** control.

20 Set this property to **true** to display the header in the  
21 **System.Web.UI.WebControls.DataGrid** . The appearance of the header is  
22 controlled by using the **System.Web.UI.WebControls.DataGrid.HeaderStyle**  
23 property.

24 Site

25 Style

1	TabIndex
2	TagKey
3	TagName
4	TemplateSourceDirectory
5	ToolTip
6	UniqueID
7	ViewState
8	ViewStateIgnoresCase
9	VirtualItemCount
10	Validate

*Description*

Gets or sets the virtual number of items in the **System.Web.UI.WebControls.DataGrid** control when custom paging is used.

Use this property to specify the virtual number of items in the **System.Web.UI.WebControls.DataGrid** control when custom paging is used.

This property is only used when the **System.Web.UI.WebControls.DataGrid.AllowCustomPaging** property is set to **true**.

Visible

Width

Validate

1  
2  
3 *Description*

4       Occurs when the **Cancel** button is clicked for an item in the  
5 **System.Web.UI.WebControls.DataGrid** control.

6       The **System.Web.UI.WebControls.DataGrid.CancelCommand** event is  
7 raised when the **Cancel** button is clicked for an item in the  
8 **System.Web.UI.WebControls.DataGrid** control.

9       Validate  
10  
11

12 *Description*

13       Occurs when the **Delete** button is clicked for an item in the  
14 **System.Web.UI.WebControls.DataGrid** control.

15       The **System.Web.UI.WebControls.DataGrid.DeleteCommand** event is  
16 raised when the **Delete** button is clicked for an item in the  
17 **System.Web.UI.WebControls.DataGrid** control.

18       Validate  
19  
20

21 *Description*

22       Occurs when the **Edit** button is clicked for an item in the  
23 **System.Web.UI.WebControls.DataGrid** control.  
24  
25

1 The **System.Web.UI.WebControls.DataGrid.EditCommand** event is  
2 raised when the **Edit** button is clicked for an item in the  
3 **System.Web.UI.WebControls.DataGrid** control.

4 Validate

5  
6  
7 *Description*

8 Occurs when any button is clicked in the  
9 **System.Web.UI.WebControls.DataGrid** control.

10 The **System.Web.UI.WebControls.DataGrid.ItemCommand** event is  
11 raised when any button is clicked in the **System.Web.UI.WebControls.DataGrid**  
12 control. This event is commonly used to handle buttons controls with a custom  
13 **CommandName** value in the **System.Web.UI.WebControls.DataGrid** control.

14 Validate

15  
16 [C#] public event DataGridItemEventHandler ItemCreated;  
17 [C++] public: \_\_event DataGridItemEventHandler\* ItemCreated;  
18 [VB] Public Event ItemCreated As DataGridItemEventHandler

19  
20 *Description*

21 Occurs on the server when an item in the  
22 **System.Web.UI.WebControls.DataGrid** control is created.

23 The **System.Web.UI.WebControls.DataGrid.ItemCreated** event is  
24 raised when an item in the **System.Web.UI.WebControls.DataGrid** control is  
25 created, both during round-trips and at data bind time.

1       Validate

2  
3   [C#] public event DataGridItemEventHandler ItemDataBound;  
4   [C++] public: \_\_event DataGridItemEventHandler\* ItemDataBound;  
5   [VB] Public Event ItemDataBound As DataGridItemEventHandler  
6

7   *Description*

8       Occurs after an item is data bound to the  
9   **System.Web.UI.WebControls.DataGrid** control.

10       The **System.Web.UI.WebControls.DataGrid.ItemDataBound** event is  
11   raised after an item is data bound to the **System.Web.UI.WebControls.DataGrid**  
12   control. This event provides you with the last opportunity to access the data item  
13   before it is displayed on the client. After this event is raised, the data item is nulled  
14   out and no longer available.

15       Validate

16  
17  
18   *Description*

19       Occurs when one of the page selection elements is clicked.

20       The **System.Web.UI.WebControls.DataGrid.PageIndexChanged** event  
21   is raised when one of the page selection elements is clicked.

22       Validate

23  
24  
25   *Description*

Occurs when a column is sorted.

The **System.Web.UI.WebControls.DataGrid.SortCommand** event is raised when a column is sorted.

Validate

### *Description*

Occurs when the **Update** button is clicked for an item in the **System.Web.UI.WebControls.DataGrid** control.

The **System.Web.UI.WebControls.DataGrid.UpdateCommand** event is raised when the **Update** button for an item is clicked.

CreateColumnSet

[C#] protected virtual ArrayList CreateColumnSet(PagedDataSource dataSource, bool useDataSource);

[C++] protected: virtual ArrayList\* CreateColumnSet(PagedDataSource\* dataSource, bool useDataSource);

[VB] Overridable Protected Function CreateColumnSet(ByVal dataSource As PagedDataSource, ByVal useDataSource As Boolean) As ArrayList

[JScript] protected function CreateColumnSet(dataSource : PagedDataSource, useDataSource : Boolean) : ArrayList;

### *Description*

Creates the set of columns to be used to build up the control hierarchy. When **AutoGenerateColumns** is true, the columns are created to match the

datasource and are appended to the set of columns defined in the Columns collection. The datasource being used to create the control hierarchy Whether to use the datasource to generate columns automatically or to use saved state.

#### CreateControlHierarchy

[C#] protected override void CreateControlHierarchy(bool useDataSource);  
[C++] protected: void CreateControlHierarchy(bool useDataSource);  
[VB] Overrides Protected Sub CreateControlHierarchy(ByVal useDataSource As Boolean)  
[JScript] protected override function CreateControlHierarchy(useDataSource : Boolean);

#### *Description*

Creates the control hierarchy that is used to render the DataGrid. This is called whenever a control hierarchy is needed and the ChildControlsCreated property is false. The implementation assumes that all the children in the controls collection have already been cleared. Whether to use the datasource to generate columns automatically or to use saved state.

#### CreateControlStyle

[C#] protected override Style CreateControlStyle();  
[C++] protected: Style\* CreateControlStyle();  
[VB] Overrides Protected Function CreateControlStyle() As Style  
[JScript] protected override function CreateControlStyle() : Style;

1  
2 *Description*

3       Creates new control style.

4 *Return Value:* A **System.Web.UI.WebControls.Style** the represents the new  
5 style.

6       CreateItem

7  
8 [C#] protected virtual DataGridItem CreateItem(int itemIndex, int  
9 dataSourceIndex, ListItemType itemType);

10 [C++] protected: virtual DataGridItem\* CreateItem(int itemIndex, int  
11 dataSourceIndex, ListItemType itemType);

12 [VB] Overridable Protected Function CreateItem(ByVal itemIndex As Integer,  
13 ByVal dataSourceIndex As Integer, ByVal itemType As ListItemType) As  
14 DataGridItem

15 [JScript] protected function CreateItem(itemIndex : int, dataSourceIndex : int,  
16 itemType : ListItemType) : DataGridItem;

17  
18 *Description*

19  
20       InitializeItem

21  
22 [C#] protected virtual void InitializeItem(DataGridItem item, DataGridColumn[]  
23 columns);

24 [C++] protected: virtual void InitializeItem(DataGridItem\* item,  
25 DataGridColumn\* columns[]);



1 [VB] Overridable Protected Sub InitializeItem(ByVal item As DataGridItem,  
2 ByVal columns() As DataGridColumn)  
3 [JScript] protected function InitializeItem(item : DataGridItem, columns :  
4 DataGridColumn[]);

5  
6 *Description*

7  
8 **InitializePager**

9  
10 [C#] protected virtual void InitializePager(DataGridItem item, int columnSpan,  
11 PagedDataSource pagedDataSource);  
12 [C++] protected: virtual void InitializePager(DataGridItem\* item, int columnSpan,  
13 PagedDataSource\* pagedDataSource);  
14 [VB] Overridable Protected Sub InitializePager(ByVal item As DataGridItem,  
15 ByVal columnSpan As Integer, ByVal pagedDataSource As PagedDataSource)  
16 [JScript] protected function InitializePager(item : DataGridItem, columnSpan : int,  
17 pagedDataSource : PagedDataSource);  
18

19 *Description*

20 Creates a DataGridItem that contains the paging UI. The paging UI is a  
21 navigation bar that is built into a single TableCell that spans across all columns  
22 of the DataGrid.

23 **LoadViewState**

24  
25 [C#] protected override void LoadViewState(object savedState);

1 [C++] protected: void LoadViewState(Object\* savedState);

2 [VB] Overrides Protected Sub LoadViewState(ByVal savedState As Object)

3 [JScript] protected override function LoadViewState(savedState : Object);

4  
5 *Description*

6 Loads a saved state of the **System.Web.UI.WebControls.DataGrid**. A  
7 saved state of the **System.Web.UI.WebControls.DataGrid**.

8 **OnBubbleEvent**

9  
10 [C#] protected override bool OnBubbleEvent(object source, EventArgs e);

11 [C++] protected: bool OnBubbleEvent(Object\* source, EventArgs\* e);

12 [VB] Overrides Protected Function OnBubbleEvent(ByVal source As Object,  
13 ByVal e As EventArgs) As Boolean

14 [JScript] protected override function OnBubbleEvent(source : Object, e :  
15 EventArgs) : Boolean;

16  
17 *Description*

18 The source of the event. An **System.EventArgs** that contains event data.

19 **OnCancelCommand**

20  
21 [C#] protected virtual void OnCancelCommand(DataGridCommandEventArgs e);

22 [C++] protected: virtual void OnCancelCommand(DataGridCommandEventArgs\*  
23 e);

24 [VB] Overridable Protected Sub OnCancelCommand(ByVal e As  
25 DataGridCommandEventArgs)

1 [JScript] protected function OnCancelCommand(e :  
2 DataGridCommandEventArgs);

3  
4 *Description*

5       Raises the **System.Web.UI.WebControls.DataGrid.CancelCommand**  
6 event. This allows you to provide a custom handler for the event.

7       Use the  
8 **System.Web.UI.WebControls.DataGrid.OnCancelCommand(System.Web.UI**  
9 **.WebControls.DataGridCommandEventArgs)** method to provide a custom  
10 handler for the **System.Web.UI.WebControls.DataGrid.CancelCommand**  
11 event. A **System.Web.UI.WebControls.DataGridCommandEventArgs** that  
12 contains event data.

13       **OnDeleteCommand**

14  
15 [C#] protected virtual void OnDeleteCommand(DataGridCommandEventArgs e);  
16 [C++] protected: virtual void OnDeleteCommand(DataGridCommandEventArgs\*  
17 e);

18 [VB] Overridable Protected Sub OnDeleteCommand(ByVal e As  
19 DataGridCommandEventArgs)

20 [JScript] protected function OnDeleteCommand(e :  
21 DataGridCommandEventArgs);

22  
23 *Description*

24       Raises the **System.Web.UI.WebControls.DataGrid.DeleteCommand**  
25 event. This allows you to provide a custom handler for the event.

1        Use the  
2        **System.Web.UI.WebControls.DataGrid.OnDeleteCommand(System.Web.UI.**  
3        **WebControls.DataGridCommandEventArgs)** method to provide a custom  
4        handler for the **System.Web.UI.WebControls.DataGrid.DeleteCommand** event.  
5        A **System.Web.UI.WebControls.DataGridCommandEventArgs** that contains  
6        event data.

#### 7        OnEditCommand

9        [C#] protected virtual void OnEditCommand(DataGridCommandEventArgs e);  
10        [C++] protected: virtual void OnEditCommand(DataGridCommandEventArgs\* e);  
11        [VB] Overridable Protected Sub OnEditCommand(ByVal e As  
12        DataGridCommandEventArgs)  
13        [JScript] protected function OnEditCommand(e : DataGridCommandEventArgs);

#### 15        *Description*

16        Raises the **System.Web.UI.WebControls.DataGrid.EditCommand**  
17        event. This allows you to provide a custom handler for the event.

18        Use the  
19        **System.Web.UI.WebControls.DataGrid.OnEditCommand(System.Web.UI.W**  
20        **ebControls.DataGridCommandEventArgs)** method to provide a custom handler  
21        for the **System.Web.UI.WebControls.DataGrid.EditCommand** event. A  
22        **System.Web.UI.WebControls.DataGridCommandEventArgs** that contains  
23        event data.

#### 24        OnItemCommand

```

1
2 [C#] protected virtual void OnItemCommand(DataGridCommandEventArgs e);
3 [C++] protected: virtual void OnItemCommand(DataGridCommandEventArgs*
4 e);
5 [VB] Overridable Protected Sub OnItemCommand(ByVal e As
6 DataGridCommandEventArgs)
7 [JScript] protected function OnItemCommand(e : DataGridCommandEventArgs);
8

```

### *Description*

Raises the **System.Web.UI.WebControls.DataGrid.ItemCommand** event. This allows you to provide a custom handler for the event. This allows you to provide a custom handler for the event.

Use the **System.Web.UI.WebControls.DataGrid.OnItemCommand(System.Web.UI.WebControls.DataGridCommandEventArgs)** method to provide a custom handler for the **System.Web.UI.WebControls.DataGrid.ItemCommand** event. A **System.Web.UI.WebControls.DataGridCommandEventArgs** that contains event data.

### **OnItemCreated**

```

21 [C#] protected virtual void OnItemCreated(DataGridItemEventArgs e);
22 [C++] protected: virtual void OnItemCreated(DataGridItemEventArgs* e);
23 [VB] Overridable Protected Sub OnItemCreated(ByVal e As
24 DataGridItemEventArgs)
25 [JScript] protected function OnItemCreated(e : DataGridItemEventArgs);

```

1  
2 *Description*

3       Raises the **System.Web.UI.WebControls.DataGrid.ItemCreated** event.  
4 This allows you to provide a custom handler for the event.

5       Use the  
6 **System.Web.UI.WebControls.DataGrid.OnItemCreated(System.Web.UI.We**  
7 **bControls.DataGridItemEventArgs)** method to provide a custom handler for the  
8 **System.Web.UI.WebControls.DataGrid.ItemCreated** event. A  
9 **System.Web.UI.WebControls.DataGridItemEventArgs** that contains event  
10 data.

11       **OnItemDataBound**

12  
13 [C#] protected virtual void OnItemDataBound(DataGridItemEventArgs e);  
14 [C++] protected: virtual void OnItemDataBound(DataGridItemEventArgs\* e);  
15 [VB] Overridable Protected Sub OnItemDataBound(ByVal e As  
16 DataGridItemEventArgs)  
17 [JScript] protected function OnItemDataBound(e : DataGridItemEventArgs);  
18

19 *Description*

20       Raises the **System.Web.UI.WebControls.DataGrid.ItemDataBound**  
21 event. This allows you to provide a custom handler for the event.

22       Use the  
23 **System.Web.UI.WebControls.DataGrid.OnItemDataBound(System.Web.UI.**  
24 **WebControls.DataGridItemEventArgs)** method to provide a custom handler for  
25 the **System.Web.UI.WebControls.DataGrid.ItemDataBound** event. A

1 **System.Web.UI.WebControls.DataGridItemEventArgs** that contains event  
2 data.

3 **OnPageIndexChanged**

4  
5 [C#] protected virtual void

6 **OnPageIndexChanged(DataGridPageChangedEventArgs e);**

7 [C++] protected: virtual void

8 **OnPageIndexChanged(DataGridPageChangedEventArgs\* e);**

9 [VB] Overridable Protected Sub **OnPageIndexChanged(ByVal e As**

10 **DataGridPageChangedEventArgs)**

11 [JScript] protected function **OnPageIndexChanged(e :**

12 **DataGridPageChangedEventArgs);**

13  
14 *Description*

15 **Raises the System.Web.UI.WebControls.DataGrid.PageIndexChanged**  
16 **event. This allows you to provide a custom handler for the event.**

17 Use the

18 **System.Web.UI.WebControls.DataGrid.OnPageIndexChanged(System.Web.**

19 **UI.WebControls.DataGridPageChangedEventArgs)** method to provide a

20 custom handler for the

21 **System.Web.UI.WebControls.DataGrid.PageIndexChanged** event. A

22 **System.Web.UI.WebControls.DataGridPageChangedEventArgs** that contains  
23 event data.

24 **OnSortCommand**

1  
2 [C#] protected virtual void OnSortCommand(DataGridSortCommandEventArgs  
3 e);

4 [C++] protected: virtual void

5 OnSortCommand(DataGridSortCommandEventArgs\* e);

6 [VB] Overridable Protected Sub OnSortCommand(ByVal e As

7 DataGridSortCommandEventArgs)

8 [JScript] protected function OnSortCommand(e :

9 DataGridSortCommandEventArgs);

10  
11 *Description*

12 Raises the **System.Web.UI.WebControls.DataGrid.SortCommand**  
13 event. This allows you to provide a custom handler for the event.

14 Use the

15 **System.Web.UI.WebControls.DataGrid.OnSortCommand(System.Web.UI.W**  
16 **ebControls.DataGridSortCommandEventArgs)** method to provide a custom  
17 handler for the **System.Web.UI.WebControls.DataGrid.SortCommand** event.

18 A **System.Web.UI.WebControls.DataGridSortCommandEventArgs** that  
19 contains event data.

20 **OnUpdateCommand**

21  
22 [C#] protected virtual void OnUpdateCommand(DataGridCommandEventArgs e);

23 [C++] protected: virtual void OnUpdateCommand(DataGridCommandEventArgs\*  
24 e);

25 [VB] Overridable Protected Sub OnUpdateCommand(ByVal e As



1 DataGridCommandEventArgs)

2 [JScript] protected function OnUpdateCommand(e :

3 DataGridCommandEventArgs);

4  
5 *Description*

6       Raises the **System.Web.UI.WebControls.DataGrid.UpdateCommand**  
7 event. This allows you to provide a custom handler for the event.

8       Use the  
9 **System.Web.UI.WebControls.DataGrid.OnUpdateCommand(System.Web.UI**  
10 **.WebControls.DataGridCommandEventArgs)** method to provide a custom  
11 handler for the **System.Web.UI.WebControls.DataGrid.UpdateCommand**  
12 event. A **System.Web.UI.WebControls.DataGridCommandEventArgs** that  
13 contains event data.

14       **PrepareControlHierarchy**

15  
16 [C#] protected override void PrepareControlHierarchy();

17 [C++] protected: void PrepareControlHierarchy();

18 [VB] Overrides Protected Sub PrepareControlHierarchy()

19 [JScript] protected override function PrepareControlHierarchy();

20  
21 *Description*

22       **SaveViewState**

23  
24 [C#] protected override object SaveViewState();

25 [C++] protected: Object\* SaveViewState();

1 [VB] Overrides Protected Function SaveViewState() As Object

2 [JScript] protected override function SaveViewState() : Object;

3  
4 *Description*

5 Saves the current state of the **System.Web.UI.WebControls.DataGrid** .

6 *Return Value:* The saved state of the **System.Web.UI.WebControls.DataGrid** .

7 TrackViewState

8  
9 [C#] protected override void TrackViewState();

10 [C++] protected: void TrackViewState();

11 [VB] Overrides Protected Sub TrackViewState()

12 [JScript] protected override function TrackViewState();

13  
14 *Description*

15 Marks the starting point to begin tracking and saving changes to the control  
16 as part of the control viewstate.

17 DataGridColumn class (System.Web.UI.WebControls)

18 TrackViewState

19  
20  
21 *Description*

22 Serves as the base class for the different column types of the  
23 **System.Web.UI.WebControls.DataGrid** control.

24 The **System.Web.UI.WebControls.DataGridColumn** class is the base  
25 class for all column types of the **System.Web.UI.WebControls.DataGrid**

control. It defines the properties and methods that are common to all column types. An instance of the **System.Web.UI.WebControls.DataGridColumn** class is typically not created directly.

**DataGridColumn**

*Example Syntax:*

**TrackViewState**

[C#] public DataGridColumn();

[C++] public: DataGridColumn();

[VB] Public Sub New()

[JScript] public function DataGridColumn();

### *Description*

Initializes a new instance of the **System.Web.UI.WebControls.DataGridColumn** class.

Use this constructor to create and initialize an instance of the **System.Web.UI.WebControls.DataGridColumn** class.

**DesignMode**

**TrackViewState**

[C#] protected bool DesignMode {get;}

[C++] protected: \_\_property bool get\_DesignMode();

[VB] Protected ReadOnly Property DesignMode As Boolean

[JScript] protected function get DesignMode() : Boolean;

1  
2 *Description*

3 Gets a value that indicates whether the column is in design mode.

4 Use the **System.Web.UI.WebControls.DataGridColumn.DesignMode**  
5 property to programmatically determine whether the column is in design mode

6 FooterStyle

7 TrackViewState

8  
9 [C#] public virtual TableItemStyle FooterStyle {get;}

10 [C++] public: \_\_property virtual TableItemStyle\* get\_FooterStyle();

11 [VB] Overridable Public ReadOnly Property FooterStyle As TableItemStyle

12 [JScript] public function get FooterStyle() : TableItemStyle;

13  
14 *Description*

15 Gets the style properties for the footer section of the column.

16 Use this property to provide a custom style for the footer section of the  
17 column. Common style attributes that can be adjusted, include forecolor,  
18 bgcolor, font, and content alignment within the cell. Providing a different style  
19 enhances the appearance of the column in the  
20 **System.Web.UI.WebControls.DataGrid** control.

21 FooterText

22 TrackViewState

23  
24 [C#] public virtual string FooterText {get; set;}

25 [C++] public: \_\_property virtual String\* get\_FooterText();public: \_\_property

1 virtual void set\_FooterText(String\*);

2 [VB] Overridable Public Property FooterText As String

3 [JScript] public function get FooterText() : String;public function set

4 FooterText(String);

5  
6 *Description*

7 Gets or sets the text displayed in the footer section of the column.

8 Use the **System.Web.UI.WebControls.DataGridColumn.FooterText**  
9 property to specify or determine the text displayed in the footer section of the  
10 column.

11 HeaderImageUrl

12 TrackViewState

13  
14 [C#] public virtual string HeaderImageUrl {get; set;}

15 [C++] public: \_\_property virtual String\* get\_HeaderImageUrl();public: \_\_property

16 virtual void set\_HeaderImageUrl(String\*);

17 [VB] Overridable Public Property HeaderImageUrl As String

18 [JScript] public function get HeaderImageUrl() : String;public function set

19 HeaderImageUrl(String);

20  
21 *Description*

22 Gets or sets the location of an image to display in the header section of the  
23 column.

24 Use the

25 **System.Web.UI.WebControls.DataGridColumn.HeaderImageUrl** property to

specify the URL of an image to display in the header section of the column. You can use a relative or an absolute URL. A relative URL relates the location of the image to the location of the Web page without specifying a complete path on the server. The path is relative to the location of the Web page. This makes it easier to move the entire site to another directory on the server without updating the path to the image in code. An absolute URL provides the complete path, so moving the site to another directory requires updating the code.

HeaderStyle

TrackViewState

[C#] public virtual TableItemStyle HeaderStyle {get;}

[C++] public: \_\_property virtual TableItemStyle\* get\_HeaderStyle();

[VB] Overridable Public ReadOnly Property HeaderStyle As TableItemStyle

[JScript] public function get HeaderStyle() : TableItemStyle;

### *Description*

Gets the style properties for the header section of the column.

Use this property to provide a custom style for the header section of the column. Common style attributes that can be adjusted include forecolor, backcolor, font, and content alignment within the cell. Providing a different style enhances the appearance of the column in the

**System.Web.UI.WebControls.DataGrid** control.

HeaderText

TrackViewState

1  
2 [C#] public virtual string HeaderText {get; set;}

3 [C++] public: \_\_property virtual String\* get\_HeaderText();public: \_\_property  
4 virtual void set\_HeaderText(String\*);

5 [VB] Overridable Public Property HeaderText As String

6 [JScript] public function get HeaderText() : String;public function set  
7 HeaderText(String);

8  
9 *Description*

10 Gets or sets the text displayed in the header section of the column.

11 Use the **System.Web.UI.WebControls.DataGridColumn.HeaderText**  
12 property to specify or determine the text displayed in the footer section of the  
13 column.

14 IsTrackingViewState

15 TrackViewState

16  
17 [C#] protected bool IsTrackingViewState {get;}

18 [C++] protected: \_\_property bool get\_IsTrackingViewState();

19 [VB] Protected ReadOnly Property IsTrackingViewState As Boolean

20 [JScript] protected function get IsTrackingViewState() : Boolean;

21  
22 *Description*

23 Determines if the **System.Web.UI.WebControls.DataGridColumn** is  
24 marked to save its state.

25 ItemStyle

## TrackViewState

```
[C#] public virtual TableItemStyle ItemStyle {get;}
[C++] public: __property virtual TableItemStyle* get_ItemStyle();
[VB] Overridable Public ReadOnly Property ItemStyle As TableItemStyle
[JScript] public function get ItemStyle() : TableItemStyle;
```

### *Description*

Gets the style properties for the item cells of the column.

Use this property to provide a custom style for the item cells of the column.

Common style attributes that can be adjusted include forecolor, backcolor, font, and content alignment within the cell. Providing a different style enhances the appearance of the column in the **System.Web.UI.WebControls.DataGrid** control.

### Owner

## TrackViewState

```
[C#] protected DataGrid Owner {get;}
[C++] protected: __property DataGrid* get_Owner();
[VB] Protected ReadOnly Property Owner As DataGrid
[JScript] protected function get Owner() : DataGrid;
```

### *Description*

Gets the **System.Web.UI.WebControls.DataGrid** control that the column is a member of.



1 Use the **System.Web.UI.WebControls.DataGridColumn.Owner**  
2 property to programmatically determine the  
3 **System.Web.UI.WebControls.DataGrid** control that the column is a member of.

4 **SortExpression**

5 **TrackViewState**

6  
7 [C#] public virtual string SortExpression {get; set;}

8 [C++] public: \_\_property virtual String\* get\_SortExpression();public: \_\_property  
9 virtual void set\_SortExpression(String\*);

10 [VB] Overridable Public Property SortExpression As String

11 [JScript] public function get SortExpression() : String;public function set  
12 SortExpression(String);

13  
14 *Description*

15 Gets or sets the name of the field to pass to the  
16 **System.Web.UI.WebControls.DataGrid.OnSortCommand(System.Web.UI.W**  
17 **ebControls.DataGridSortCommandEventArgs)** method when a column is  
18 selected for sorting.

19 Use the  
20 **System.Web.UI.WebControls.DataGridColumn.SortExpression** property to  
21 specify or determine the name of the field to pass to the  
22 **System.Web.UI.WebControls.DataGrid.OnSortCommand(System.Web.UI.W**  
23 **ebControls.DataGridSortCommandEventArgs)** method when a column is  
24 selected for sorting.

25 **ViewState**

## TrackViewState

[C#] protected StateBag ViewState {get;}

[C++] protected: \_\_property StateBag\* get\_ViewState();

[VB] Protected ReadOnly Property ViewState As StateBag

[JScript] protected function get ViewState() : StateBag;

### *Description*

Gets the statebag for the **System.Web.UI.WebControls.DataGridColumn**.

This property is read-only.

Visible

TrackViewState

[C#] public bool Visible {get; set;}

[C++] public: \_\_property bool get\_Visible();public: \_\_property void set\_Visible(bool);

[VB] Public Property Visible As Boolean

[JScript] public function get Visible() : Boolean;public function set Visible(Boolean);

### *Description*

Gets or sets a value that indicates whether the column is visible in the **System.Web.UI.WebControls.DataGrid** control.

1 Use the **System.Web.UI.WebControls.DataGridColumn.Visible**  
2 property to programmatically control whether the column is visible in the  
3 **System.Web.UI.WebControls.DataGrid** control.

#### 4 Initialize

5  
6 [C#] public virtual void Initialize();  
7 [C++] public: virtual void Initialize();  
8 [VB] Overridable Public Sub Initialize()  
9 [JScript] public function Initialize();

#### 10 11 *Description*

#### 12 InitializeCell

13  
14 [C#] public virtual void InitializeCell(TableCell cell, int columnIndex,  
15 ListItemType itemType);  
16 [C++] public: virtual void InitializeCell(TableCell\* cell, int columnIndex,  
17 ListItemType itemType);  
18 [VB] Overridable Public Sub InitializeCell(ByVal cell As TableCell, ByVal  
19 columnIndex As Integer, ByVal itemType As ListItemType)  
20 [JScript] public function InitializeCell(cell : TableCell, columnIndex : int,  
21 itemType : ListItemType);

#### 22 23 *Description*

24 Initializes a cell in the **System.Web.UI.WebControls.DataGridColumn** .  
25 A **System.Web.UI.WebControls.TableCell** that contains information about the

cell. The column number of the cell to initialize. One of the

**System.Web.UI.WebControls.ListItemType** values.

**LoadViewState**

[C#] protected virtual void LoadViewState(object savedState);

[C++] protected: virtual void LoadViewState(Object\* savedState);

[VB] Overridable Protected Sub LoadViewState(ByVal savedState As Object)

[JScript] protected function LoadViewState(savedState : Object);

*Description*

Loads the state of the **System.Web.UI.WebControls.DataGridColumn**.

An **System.Object** that contains the saved state of the

**System.Web.UI.WebControls.DataGridColumn**.

**OnColumnChanged**

[C#] protected virtual void OnColumnChanged();

[C++] protected: virtual void OnColumnChanged();

[VB] Overridable Protected Sub OnColumnChanged()

[JScript] protected function OnColumnChanged();

*Description*

Raises the **ColumnChanged** event of a

**System.Web.UI.WebControls.DataGridColumn** object.

**SaveViewState**

1  
2 [C#] protected virtual object SaveViewState();

3 [C++] protected: virtual Object\* SaveViewState();

4 [VB] Overridable Protected Function SaveViewState() As Object

5 [JScript] protected function SaveViewState() : Object;

6  
7 *Description*

8 Saves the current state of the

9 **System.Web.UI.WebControls.DataGridColumn .**

10 *Return Value:* An **System.Object** that contains the saved state of the

11 **System.Web.UI.WebControls.DataGridColumn .**

12 IStateManager.LoadViewState

13  
14 [C#] void IStateManager.LoadViewState(object state);

15 [C++] void IStateManager::LoadViewState(Object\* state);

16 [VB] Sub LoadViewState(ByVal state As Object) Implements

17 IStateManager.LoadViewState

18 [JScript] function IStateManager.LoadViewState(state : Object);

19 IStateManager.SaveViewState

20  
21 [C#] object IStateManager.SaveViewState();

22 [C++] Object\* IStateManager::SaveViewState();

23 [VB] Function SaveViewState() As Object Implements

24 IStateManager.SaveViewState

25 [JScript] function IStateManager.SaveViewState() : Object;

1            IStateManager.TrackViewState  
2  
3    [C#] void IStateManager.TrackViewState();  
4    [C++] void IStateManager::TrackViewState();  
5    [VB] Sub TrackViewState() Implements IStateManager.TrackViewState  
6    [JScript] function IStateManager.TrackViewState();

7            ToString

8  
9    [C#] public override string ToString();  
10    [C++] public: String\* ToString();  
11    [VB] Overrides Public Function ToString() As String  
12    [JScript] public override function ToString() : String;

13  
14    *Description*

15            Returns the string representation of the column.

16    *Return Value:* Returns **System.String.Empty** .

17            Use the **System.Web.UI.WebControls.DataGridColumn.ToString**  
18    method to get the string representation of the column.

19            TrackViewState

20  
21    [C#] protected virtual void TrackViewState();  
22    [C++] protected: virtual void TrackViewState();  
23    [VB] Overridable Protected Sub TrackViewState()  
24    [JScript] protected function TrackViewState();  
25

1  
2 *Description*

3 Marks the starting point to begin tracking and saving changes to the control  
4 as part of the control viewstate.

5 DataGridColumnCollection class (System.Web.UI.WebControls)

6 TrackViewState

7  
8  
9 *Description*

10 A collection of **System.Web.UI.WebControls.DataGridColumn** derived  
11 column objects that represent the columns in a  
12 **System.Web.UI.WebControls.DataGrid** control. This class cannot be inherited.

13 Use the **System.Web.UI.WebControls.DataGridColumnCollection** to  
14 programmatically manage a collection of  
15 **System.Web.UI.WebControls.DataGridColumn** derived column objects. These  
16 objects represent the columns in a **System.Web.UI.WebControls.DataGrid**  
17 control. You can add, remove, or insert columns into the  
18 **System.Web.UI.WebControls.DataGridColumnCollection** .

19 DataGridColumnCollection

20 *Example Syntax:*

21 TrackViewState

22  
23 [C#] public DataGridColumnCollection(DataGrid owner, ArrayList columns);

24 [C++] public: DataGridColumnCollection(DataGrid\* owner, ArrayList\*  
25 columns);

```
1 [VB] Public Sub New(ByVal owner As DataGrid, ByVal columns As ArrayList)
2 [JScript] public function DataGridColumnCollection(owner : DataGrid, columns :
3 ArrayList);
```

#### *Description*

Initializes a new instance of the  
**System.Web.UI.WebControls.DataGridColumnCollection** class.

Use this constructor to create and initialize a new instance of the  
**System.Web.UI.WebControls.DataGridColumnCollection** class. The  
**System.Web.UI.WebControls.DataGrid** control that corresponds with this  
collection. A **System.Collections.ArrayList** object that stores the collection of  
columns.

Count

TrackViewState

```
16 [C#] public int Count {get;}
17 [C++] public: __property int get_Count();
18 [VB] Public ReadOnly Property Count As Integer
19 [JScript] public function get Count() : int;
```

#### *Description*

Gets the number of columns in the  
**System.Web.UI.WebControls.DataGridColumnCollection** .

Use this property to determine the number of columns in the  
**System.Web.UI.WebControls.DataGridColumnCollection** . The



**System.Web.UI.WebControls.DataGridColumnCollection.Count** property is commonly used when iterating through the collection to determine the upper bound of the collection.

IsReadOnly

TrackViewState

[C#] public bool IsReadOnly {get;}

[C++] public: \_\_property bool get\_IsReadOnly();

[VB] Public ReadOnly Property IsReadOnly As Boolean

[JScript] public function get IsReadOnly() : Boolean;

#### *Description*

Gets a value that indicates whether the columns in the **System.Web.UI.WebControls.DataGridColumnCollection** can be modified.

This property always returns **false** to indicate that the **System.Web.UI.WebControls.DataGridColumnCollection** can be written to in all cases.

IsSynchronized

TrackViewState

[C#] public bool IsSynchronized {get;}

[C++] public: \_\_property bool get\_IsSynchronized();

[VB] Public ReadOnly Property IsSynchronized As Boolean

[JScript] public function get IsSynchronized() : Boolean;

1  
2 *Description*

3 Gets a value indicating whether access to the  
4 **System.Web.UI.WebControls.DataGridColumnCollection** is synchronized  
5 (thread-safe).

6 This property is derived from the **System.Collections.ICollection** class and  
7 is overridden to always return **false** .

8 Item

9 TrackViewState

10  
11 [C#] public DataGridColumn this[int index] {get;}

12 [C++] public: \_\_property DataGridColumn\* get\_Item(int index);

13 [VB] Public Default ReadOnly Property Item(ByVal index As Integer) As

14 DataGridColumn

15 [JScript] returnValue = DataGridColumnCollectionObject.Item(index);

16  
17 *Description*

18 Gets a **System.Web.UI.WebControls.DataGridColumn** derived column  
19 object from the **System.Web.UI.WebControls.DataGridColumnCollection**  
20 collection at the specified index.

21 Use this indexer to get a **System.Web.UI.WebControls.DataGridColumn**  
22 derived column object from the

23 **System.Web.UI.WebControls.DataGridColumnCollection** at the specified  
24 index using array notation. The index of the

25 **System.Web.UI.WebControls.DataGridColumn** derived object in the

1 **System.Web.UI.WebControls.DataGridColumnCollection** collection to  
2 retrieve.

3       SyncRoot

4       TrackViewState

6 [C#] public object SyncRoot {get;}

7 [C++] public: \_\_property Object\* get\_SyncRoot();

8 [VB] Public ReadOnly Property SyncRoot As Object

9 [JScript] public function get SyncRoot() : Object;

11 *Description*

12       Gets the object that can be used to synchronize access to the  
13 **System.Web.UI.WebControls.DataGridColumnCollection** .

14       The object returned in this implementation is the  
15 **System.Web.UI.WebControls.DataGridColumnCollection** object.

16       Add

18 [C#] public void Add(DataGridColumn column);

19 [C++] public: void Add(DataGridColumn\* column);

20 [VB] Public Sub Add(ByVal column As DataGridColumn)

21 [JScript] public function Add(column : DataGridColumn);

23 *Description*

24

25

1 Appends the specified **System.Web.UI.WebControls.DataGridColumn**  
2 derived column object to the end of the  
3 **System.Web.UI.WebControls.DataGridColumnCollection** .

4 Use this method to add a  
5 **System.Web.UI.WebControls.DataGridColumn** derived column object to the  
6 end of a **System.Web.UI.WebControls.DataGridColumnCollection** . The  
7 **System.Web.UI.WebControls.DataGridColumn** derived column object to  
8 append to the **System.Web.UI.WebControls.DataGridColumnCollection**.

#### 9 AddAt

10  
11 [C#] public void AddAt(int index, DataGridColumn column);  
12 [C++] public: void AddAt(int index, DataGridColumn\* column);  
13 [VB] Public Sub AddAt(ByVal index As Integer, ByVal column As  
14 DataGridColumn)  
15 [JScript] public function AddAt(index : int, column : DataGridColumn);  
16

#### 17 *Description*

18 Inserts a **System.Web.UI.WebControls.DataGridColumn** derived  
19 column object in the **System.Web.UI.WebControls.DataGridColumnCollection**  
20 at the specified index.

21 Use this method to insert a  
22 **System.Web.UI.WebControls.DataGridColumn** derived column object at the  
23 specified index location in the  
24 **System.Web.UI.WebControls.DataGridColumnCollection** . The index location  
25 in the **System.Web.UI.WebControls.DataGridColumnCollection** to insert the

1 **System.Web.UI.WebControls.DataGridColumn** derived column object. The  
2 **System.Web.UI.WebControls.DataGridColumn** derived column object to insert  
3 into the **System.Web.UI.WebControls.DataGridColumnCollection**.

4 Clear

6 [C#] public void Clear();

7 [C++] public: void Clear();

8 [VB] Public Sub Clear()

9 [JScript] public function Clear();

11 *Description*

12 Removes all **System.Web.UI.WebControls.DataGridColumn** derived  
13 column objects from the

14 **System.Web.UI.WebControls.DataGridColumnCollection** .

15 Use this method to remove all

16 **System.Web.UI.WebControls.DataGridColumn** derived column objects from  
17 the **System.Web.UI.WebControls.DataGridColumnCollection** .

18 CopyTo

20 [C#] public void CopyTo(Array array, int index);

21 [C++] public: \_\_sealed void CopyTo(Array\* array, int index);

22 [VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As  
23 Integer)

24 [JScript] public function CopyTo(array : Array, index : int);

## Description

Copies the items from the **System.Web.UI.WebControls.DataGridColumnCollection** to the specified **System.Array** object, starting at the specified index in the **System.Array** object.

Use this method to copy the contents of the **System.Web.UI.WebControls.DataGridColumnCollection** into the specified **System.Array** object, starting at the specified index. A zero-based **System.Array** object that receives the copied items from the **System.Web.UI.WebControls.DataGridColumnCollection**. The first position in the specified **System.Array** object to receive the copied contents.

## GetEnumerator

[C#] public IEnumerator GetEnumerator();

[C++] public: \_\_sealed IEnumerator\* GetEnumerator();

[VB] NotOverridable Public Function GetEnumerator() As IEnumerator

[JScript] public function GetEnumerator() : IEnumerator;

## Description

Returns an **System.Collections.IEnumerator** interface that contains all **System.Web.UI.WebControls.DataGridColumn** derived column objects in the **System.Web.UI.WebControls.DataGridColumnCollection**.

*Return Value:* A **System.Collections.IEnumerator** interface that contains all **System.Web.UI.WebControls.DataGridColumn** derived column objects in the **System.Web.UI.WebControls.DataGridColumnCollection**.

Use this method to create a **System.Collections.IEnumerator** that can be iterated through easily to get each item in the

**System.Web.UI.WebControls.DataGridColumnCollection** .

**IndexOf**

[C#] public int IndexOf(DataGridColumn column);

[C++] public: int IndexOf(DataGridColumn\* column);

[VB] Public Function IndexOf(ByVal column As DataGridColumn) As Integer

[JScript] public function IndexOf(column : DataGridColumn) : int;

### *Description*

Returns the index of the specified

**System.Web.UI.WebControls.DataGridColumn** derived column object from the **System.Web.UI.WebControls.DataGridColumnCollection** .

*Return Value:* The index position of the specified

**System.Web.UI.WebControls.DataGridColumn** derived column object in the **System.Web.UI.WebControls.DataGridColumnCollection** . The default value is **-1** , which indicates that the specified

**System.Web.UI.WebControls.DataGridColumn** derived object is not found.

Use this method to determine the index number of the specified

**System.Web.UI.WebControls.DataGridColumn** derived column object in the **System.Web.UI.WebControls.DataGridColumnCollection** . If the specified **System.Web.UI.WebControls.DataGridColumn** derived column object is not found, an index of **-1** is returned. The

1 **System.Web.UI.WebControls.DataGridColumn** derived column object to  
2 search for in the **System.Web.UI.WebControls.DataGridColumnCollection**.

3 Remove

4  
5 [C#] public void Remove(DataGridColumn column);  
6 [C++] public: void Remove(DataGridColumn\* column);  
7 [VB] Public Sub Remove(ByVal column As DataGridColumn)  
8 [JScript] public function Remove(column : DataGridColumn);  
9

10 *Description*

11 Removes the specified **System.Web.UI.WebControls.DataGridColumn**  
12 derived column object from the  
13 **System.Web.UI.WebControls.DataGridColumnCollection** .

14 Use this method to remove the specified  
15 **System.Web.UI.WebControls.DataGridColumn** derived column object from a  
16 **System.Web.UI.WebControls.DataGridColumnCollection** . The  
17 **System.Web.UI.WebControls.DataGridColumn** derived column object to  
18 remove from the **System.Web.UI.WebControls.DataGridColumnCollection**.

19 RemoveAt

20  
21 [C#] public void RemoveAt(int index);  
22 [C++] public: void RemoveAt(int index);  
23 [VB] Public Sub RemoveAt(ByVal index As Integer)  
24 [JScript] public function RemoveAt(index : int);  
25



1  
2 *Description*

3 Removes a **System.Web.UI.WebControls.DataGridColumn** derived  
4 column object from the  
5 **System.Web.UI.WebControls.DataGridColumnCollection** at the specified  
6 index.

7 Use this method to remove a  
8 **System.Web.UI.WebControls.DataGridColumn** derived column object from a  
9 **System.Web.UI.WebControls.DataGridColumnCollection** at the specified  
10 index. The index of the **System.Web.UI.WebControls.DataGridColumn**  
11 derived column object in the  
12 **System.Web.UI.WebControls.DataGridColumnCollection** to remove.

13 **StateManager.LoadViewState**

14  
15 [C#] void **StateManager.LoadViewState**(object savedState);

16 [C++] void **StateManager::LoadViewState**(Object\* savedState);

17 [VB] Sub **LoadViewState**(ByVal savedState As Object) Implements  
18 **StateManager.LoadViewState**

19 [JScript] function **StateManager.LoadViewState**(savedState : Object);

20 **StateManager.SaveViewState**

21  
22 [C#] object **StateManager.SaveViewState**();

23 [C++] Object\* **StateManager::SaveViewState**();

24 [VB] Function **SaveViewState**() As Object Implements  
25

1 IStateManager.SaveViewState

2 [JScript] function IStateManager.SaveViewState() : Object;

3 IStateManager.TrackViewState

5 [C#] void IStateManager.TrackViewState();

6 [C++] void IStateManager::TrackViewState();

7 [VB] Sub TrackViewState() Implements IStateManager.TrackViewState

8 [JScript] function IStateManager.TrackViewState();

9 DataGridCommandEventArgs class (System.Web.UI.WebControls)

10 ToString

### 13 *Description*

14 Provides data for the

15 **System.Web.UI.WebControls.DataGrid.CancelCommand** ,

16 **System.Web.UI.WebControls.DataGrid.DeleteCommand** ,

17 **System.Web.UI.WebControls.DataGrid.EditCommand** ,

18 **System.Web.UI.WebControls.DataGrid.ItemCommand** , and

19 **System.Web.UI.WebControls.DataGrid.UpdateCommand** events of the

20 **System.Web.UI.WebControls.DataGrid** control. This class cannot be inherited.

21 The **System.Web.UI.WebControls.DataGrid.CancelCommand** event is  
22 raised when the **Cancel** button for an item in the

23 **System.Web.UI.WebControls.DataGrid** control is clicked.

24 DataGridCommandEventArgs

25 *Example Syntax:*

ToString

```
[C#] public DataGridCommandEventArgs(DataGridItem item, object  
commandSource, CommandEventArgs originalArgs);  
[C++] public: DataGridCommandEventArgs(DataGridItem* item, Object*  
commandSource, CommandEventArgs* originalArgs);  
[VB] Public Sub New(ByVal item As DataGridItem, ByVal commandSource As  
Object, ByVal originalArgs As CommandEventArgs)  
[JScript] public function DataGridCommandEventArgs(item : DataGridItem,  
commandSource : Object, originalArgs : CommandEventArgs);
```

### *Description*

Initializes a new instance of the

**System.Web.UI.WebControls.DataGridCommandEventArgs** class.

Use this constructor to create and initialize a new instance of the

**System.Web.UI.WebControls.DataGridCommandEventArgs** class. A

**System.Web.UI.WebControls.DataGridItem** that represents the selected item in

the **System.Web.UI.WebControls.DataGrid**. The source of the command. A

**System.Web.UI.WebControls.CommandEventArgs** that contains the event  
data.

CommandArgument

CommandName

CommandSource

ToString

1  
2  
3 *Description*

4 Gets the source of the command.

5 Use the

6 **System.Web.UI.WebControls.DataGridCommandEventArgs.CommandSource**  
7 property to determine the command source that raised the event. This property  
8 is commonly used to determine which command raised the event.

9 Item

10 ToString

11  
12 [C#] public DataGridItem Item {get;}

13 [C++] public: \_\_property DataGridItem\* get\_Item();

14 [VB] Public ReadOnly Property Item As DataGridItem

15 [JScript] public function get Item() : DataGridItem;

16  
17 *Description*

18 Gets the item containing the command source in the

19 **System.Web.UI.WebControls.DataGrid** control.

20 The **System.Web.UI.WebControls.DataGridCommandEventArgs.Item**  
21 property is used to access the properties of the selected item in the

22 **System.Web.UI.WebControls.DataGrid** control.

23 DataGridCommandEventHandler delegate (System.Web.UI.WebControls)

24 ToString

1  
2  
3 *Description*

4 Represents the method that will handle the  
5 **System.Web.UI.WebControls.DataGrid.CancelCommand** ,  
6 **System.Web.UI.WebControls.DataGrid.DeleteCommand** ,  
7 **System.Web.UI.WebControls.DataGrid.EditCommand** ,  
8 **System.Web.UI.WebControls.DataGrid.ItemCommand** , and  
9 **System.Web.UI.WebControls.DataGrid.UpdateCommand** events of a  
10 **System.Web.UI.WebControls.DataGrid** . The source of the event. A  
11 **System.Web.UI.WebControls.DataGridCommandEventArgs** that contains the  
12 event data.

13 When you create a  
14 **System.Web.UI.WebControls.DataGridCommandEventHandler** delegate, you  
15 identify the method that will handle the event. To associate the event with your  
16 event handler, add an instance of the delegate to the event. The event handler is  
17 called whenever the event occurs, unless you remove the delegate. For more  
18 information about event handler delegates, see .

19 **DataGridItem** class (System.Web.UI.WebControls)

20 **ToString**

21  
22  
23 *Description*

24 Represents an item (row) in the **System.Web.UI.WebControls.DataGrid**  
25 control.

1 A **System.Web.UI.WebControls.DataGridItem** object represents an item  
2 (row) in the **System.Web.UI.WebControls.DataGrid** control, such as the  
3 heading section, the footer section, or a data row.

4 **DataGridItem**

5 *Example Syntax:*

6 **ToString**

7  
8 [C#] public **DataGridItem**(int itemIndex, int dataSetIndex, ListItemType  
9 itemType);

10 [C++] public: **DataGridItem**(int itemIndex, int dataSetIndex, ListItemType  
11 itemType);

12 [VB] Public Sub New(ByVal itemIndex As Integer, ByVal dataSetIndex As  
13 Integer, ByVal itemType As ListItemType)

14 [JScript] public function **DataGridItem**(itemIndex : int, dataSetIndex : int,  
15 itemType : ListItemType);

16  
17 *Description*

18 Initializes a new instance of the  
19 **System.Web.UI.WebControls.DataGridItem** class.

20 Use this constructor to create and initialize a new instance of the  
21 **System.Web.UI.WebControls.DataGridItem** class. The index of the item in the  
22 **System.Web.UI.WebControls.DataGrid** control from the  
23 **System.Web.UI.WebControls.DataGrid.Items** collection. The index number of  
24 the item, from the bound data source, that appears in the  
25

**System.Web.UI.WebControls.DataGrid** control. One of the  
**System.Web.UI.WebControls.ListItemType** values.

AccessKey

Attributes

BackColor

BorderColor

BorderStyle

BorderWidth

Cells

ChildControlsCreated

ClientID

Context

Controls

ControlStyle

ControlStyleCreated

CssClass

DataItem

ToString

*Description*

Gets or sets the data item represented by the  
**System.Web.UI.WebControls.DataGridItem** object in the  
**System.Web.UI.WebControls.DataGrid** control.

1        Use the **System.Web.UI.WebControls.DataGridItem.DataItem** property  
2 to specify or determine the properties of a data item represented by the  
3 **System.Web.UI.WebControls.DataGridItem** object in the  
4 **System.Web.UI.WebControls.DataGrid** control.

5        DataSetIndex

6        ToString

7  
8 [C#] public virtual int DataSetIndex {get;}

9 [C++] public: \_\_property virtual int get\_DataSetIndex();

10 [VB] Overridable Public ReadOnly Property DataSetIndex As Integer

11 [JScript] public function get DataSetIndex() : int;

12  
13 *Description*

14        Gets the index number the **System.Web.UI.WebControls.DataGridItem**  
15 object from the bound data source.

16        Use the **System.Web.UI.WebControls.DataGridItem.DataSetIndex**  
17 property to the index number of the  
18 **System.Web.UI.WebControls.DataGridItem** object from the bound data source.

19        Enabled

20        EnableViewState

21        Events

22        Font

23        ForeColor

24        HasChildViewState

25        Height



1 HorizontalAlign  
2 ID  
3 IsTrackingViewState  
4 ItemIndex  
5 ToString

6  
7  
8 *Description*

9 Gets the index of the **System.Web.UI.WebControls.DataGridItem** object  
10 from the **System.Web.UI.WebControls.DataGrid.Items** collection of the  
11 **System.Web.UI.WebControls.DataGrid** control.

12 Use the **System.Web.UI.WebControls.DataGridItem.ItemIndex**  
13 property to determine the index number of the  
14 **System.Web.UI.WebControls.DataGridItem** object from the  
15 **System.Web.UI.WebControls.DataGrid.Items** collection of the  
16 **System.Web.UI.WebControls.DataGrid** control.

17 ItemType  
18 ToString

19  
20 [C#] public virtual ListItemType ItemType {get;}  
21 [C++] public: \_\_property virtual ListItemType get\_ItemType();  
22 [VB] Overridable Public ReadOnly Property ItemType As ListItemType  
23 [JScript] public function get ItemType() : ListItemType;

24  
25 *Description*

1 Gets the type of the item represented by the  
2 **System.Web.UI.WebControls.DataGridItem** object in the  
3 **System.Web.UI.WebControls.DataGrid** control.

4 Use the **System.Web.UI.WebControls.DataGridItem.ItemType** property  
5 to determine the type of an item in the **System.Web.UI.WebControls.DataGrid**  
6 control. The following table lists the various item types.

7 NamingContainer  
8 Page  
9 Parent  
10 Site  
11 Style  
12 TabIndex  
13 TagKey  
14 TagName  
15 TemplateSourceDirectory  
16 ToolTip  
17 UniqueID  
18 VerticalAlign  
19 ViewState  
20 ViewStateIgnoresCase  
21 Visible  
22 Width  
23 OnBubbleEvent

24  
25 [C#] protected override bool OnBubbleEvent(object source, EventArgs e);

1 [C++] protected: bool OnBubbleEvent(Object\* source, EventArgs\* e);  
 2 [VB] Overrides Protected Function OnBubbleEvent(ByVal source As Object,  
 3 ByVal e As EventArgs) As Boolean  
 4 [JScript] protected override function OnBubbleEvent(source : Object, e :  
 5 EventArgs) : Boolean;

7 *Description*

8       SetItemType

10 [C#] protected internal virtual void SetItemType(ListItemType itemType);  
 11 [C++] protected public: virtual void SetItemType(ListItemType itemType);  
 12 [VB] Overridable Protected Friend Dim Sub SetItemType(ByVal itemType As  
 13 ListItemType)  
 14 [JScript] package function SetItemType(itemType : ListItemType);

16 *Description*

17       DataGridItemCollection class (System.Web.UI.WebControls)

18       TrackViewState

21 *Description*

22       Represents a collection of **System.Web.UI.WebControls.DataGridItem**  
 23 objects in a **System.Web.UI.WebControls.DataGrid** control.

24       The **System.Web.UI.WebControls.DataGridItemCollection** class  
 25 represents a collection of **System.Web.UI.WebControls.DataGridItem** objects,

which in turn represent the data items in a

**System.Web.UI.WebControls.DataGrid** control. To programmatically retrieve

**System.Web.UI.WebControls.DataGridItem** objects from a

**System.Web.UI.WebControls.DataGrid** control, use one of following methods:

Use the indexer to get a single **System.Web.UI.WebControls.DataGridItem**

object from the collection, using array notation.

**DataGridItemCollection**

*Example Syntax:*

**TrackViewState**

[C#] public DataGridItemCollection(ArrayList items);

[C++] public: DataGridItemCollection(ArrayList\* items);

[VB] Public Sub New(ByVal items As ArrayList)

[JScript] public function DataGridItemCollection(items : ArrayList);

### *Description*

Initializes a new instance of the

**System.Web.UI.WebControls.DataGridItemCollection** class.

Use this constructor to create and initialize a new instance of the

**System.Web.UI.WebControls.DataGridItemCollection** class. A

**System.Collections.ArrayList** object that contains the items with which to

initialize the collection.

**Count**

**TrackViewState**

1  
2 [C#] public int Count {get;}

3 [C++] public: \_\_property int get\_Count();

4 [VB] Public ReadOnly Property Count As Integer

5 [JScript] public function get Count() : int;

6  
7 *Description*

8 Gets the number of **System.Web.UI.WebControls.DataGridItem** objects  
9 in the collection.

10 Use the **System.Web.UI.WebControls.DataGridItemCollection.Count**  
11 property to determine the number of  
12 **System.Web.UI.WebControls.DataGridItem** objects in the  
13 **System.Web.UI.WebControls.DataGridItemCollection** collection. The  
14 **System.Web.UI.WebControls.DataGridItemCollection.Count** property is  
15 commonly used when iterating through the collection to determine the upper  
16 bound of the collection.

17 **IsReadOnly**

18 **TrackViewState**

19  
20 [C#] public bool IsReadOnly {get;}

21 [C++] public: \_\_property bool get\_IsReadOnly();

22 [VB] Public ReadOnly Property IsReadOnly As Boolean

23 [JScript] public function get IsReadOnly() : Boolean;

24  
25 *Description*

1 Gets a value that indicates whether the  
2 **System.Web.UI.WebControls.DataGridItem** objects in the  
3 **System.Web.UI.WebControls.DataGridItemCollection** can be modified.

4 This property always returns **false** to indicate that the  
5 **System.Web.UI.WebControls.DataGridItemCollection** can be written to in all  
6 cases.

7 IsSynchronized

8 TrackViewState

9  
10 [C#] public bool IsSynchronized {get;}

11 [C++] public: \_\_property bool get\_IsSynchronized();

12 [VB] Public ReadOnly Property IsSynchronized As Boolean

13 [JScript] public function get IsSynchronized() : Boolean;

14  
15 *Description*

16 Gets a value indicating whether access to the  
17 **System.Web.UI.WebControls.DataGridItemCollection** is synchronized  
18 (thread-safe).

19 This property is derived from the **System.Collections.ICollection** class and  
20 is overridden to always return **false** .

21 Item

22 TrackViewState

23  
24 [C#] public DataGridItem this[int index] {get;}

25 [C++] public: \_\_property DataGridItem\* get\_Item(int index);

1 [VB] Public Default ReadOnly Property Item(ByVal index As Integer) As

2 DataGridItem

3 [JScript] returnValue = DataGridItemCollectionObject.Item(index);

4  
5 *Description*

6 Gets the **System.Web.UI.WebControls.DataGridItem** object at the  
7 specified index in the collection.

8 Use this indexer to get a **System.Web.UI.WebControls.DataGridItem**  
9 object from the **System.Web.UI.WebControls.DataGridItemCollection** at the  
10 specified index, using array notation. The zero-based index of the  
11 **System.Web.UI.WebControls.DataGridItem** object to retrieve from the  
12 collection.

13 SyncRoot

14 TrackViewState

15  
16 [C#] public object SyncRoot {get;}

17 [C++] public: \_\_property Object\* get\_SyncRoot();

18 [VB] Public ReadOnly Property SyncRoot As Object

19 [JScript] public function get SyncRoot() : Object;

20  
21 *Description*

22 Gets the object that can be used to synchronize access to the  
23 **System.Web.UI.WebControls.DataGridItemCollection** .

24 The object returned in this implementation is the  
25 **System.Web.UI.WebControls.DataGridItemCollection** object itself.

## CopyTo

[C#] public void CopyTo(Array array, int index);

[C++] public: \_\_sealed void CopyTo(Array\* array, int index);

[VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As Integer)

[JScript] public function CopyTo(array : Array, index : int);

### *Description*

Copies all the items from this **System.Web.UI.WebControls.DataGridItemCollection** to the specified **System.Array** object, starting at the specified index in the **System.Array** object.

Use this method to copy the contents of the **System.Web.UI.WebControls.DataGridItemCollection** collection into the specified **System.Array** object, starting at the specified index. A zero-based **System.Array** object that receives the copied items from the **System.Web.UI.WebControls.DataGridItemCollection**. The first position in the specified **System.Array** object to receive the copied contents.

### GetEnumerator

[C#] public IEnumerator GetEnumerator();

[C++] public: \_\_sealed IEnumerator\* GetEnumerator();

[VB] NotOverridable Public Function GetEnumerator() As IEnumerator

[JScript] public function GetEnumerator() : IEnumerator;



1  
2 *Description*

3 Returns a **System.Collections.IEnumerator** interface that contains all  
4 **System.Web.UI.WebControls.DataGridItem** objects in the  
5 **System.Web.UI.WebControls.DataGridItemCollection** .

6 *Return Value:* A **System.Collections.IEnumerator** interface that contains all  
7 **System.Web.UI.WebControls.DataGridItem** objects in the  
8 **System.Web.UI.WebControls.DataGridItemCollection** .

9 Use this method to create a **System.Collections.IEnumerator** that can be  
10 easily iterated through to get each item in the  
11 **System.Web.UI.WebControls.DataGridItemCollection** .

12 DataGridItemEventArgs class (System.Web.UI.WebControls)

13 ToString  
14  
15

16 *Description*

17 Provides data for the  
18 **System.Web.UI.WebControls.DataGrid.ItemCreated** and  
19 **System.Web.UI.WebControls.DataGrid.ItemDataBound** events of the  
20 **System.Web.UI.WebControls.DataGrid** control. This class cannot be inherited.

21 The **System.Web.UI.WebControls.DataGrid.ItemCreated** event is  
22 raised when an item in the **System.Web.UI.WebControls.DataGrid** control is  
23 created.

24 DataGridItemEventArgs

25 *Example Syntax:*

ToString

```
[C#] public DataGridItemEventArgs(DataGridItem item);  
[C++] public: DataGridItemEventArgs(DataGridItem* item);  
[VB] Public Sub New(ByVal item As DataGridItem)  
[JScript] public function DataGridItemEventArgs(item : DataGridItem);
```

*Description*

Initializes a new instance of  
**System.Web.UI.WebControls.DataGridItemEventArgs** class.

Use this constructor to create and initialize a new instance of the  
**System.Web.UI.WebControls.DataGridItemEventArgs** class. A  
**System.Web.UI.WebControls.DataGridItem** that represents an item in the  
**System.Web.UI.WebControls.DataGrid**.

Item

ToString

```
[C#] public DataGridItem Item {get;}  
[C++] public: __property DataGridItem* get_Item();  
[VB] Public ReadOnly Property Item As DataGridItem  
[JScript] public function get Item() : DataGridItem;
```

*Description*

Gets the referenced item in the **System.Web.UI.WebControls.DataGrid**  
control when the event is raised.

1 Use this property to programmatically access the item referenced in the  
2 **System.Web.UI.WebControls.DataGrid** control when the event is raised.

3 DataGridItemEventHandler delegate (System.Web.UI.WebControls)

4 ToString

5  
6  
7 *Description*

8 Represents the method that will handle the  
9 **System.Web.UI.WebControls.DataGrid.ItemCreated** and  
10 **System.Web.UI.WebControls.DataGrid.ItemDataBound** events of a  
11 **System.Web.UI.WebControls.DataGrid** . The source of the event. A  
12 **System.Web.UI.WebControls.DataGridItemEventArgs** than contains the event  
13 data.

14 The **System.Web.UI.WebControls.DataGrid.ItemCreated** event is  
15 raised when an item in the **System.Web.UI.WebControls.DataGrid** control is  
16 created.

17 DataGridPageChangedEventArgs class (System.Web.UI.WebControls)

18 ToString

19  
20  
21 *Description*

22 Provides data for the  
23 **System.Web.UI.WebControls.DataGrid.PageIndexChanged** event of the  
24 **System.Web.UI.WebControls.DataGrid** control. This class cannot be inherited.

The **System.Web.UI.WebControls.DataGrid.PageIndexChanged** event is raised when a button in the page selection element of the **System.Web.UI.WebControls.DataGrid** control is clicked.

**DataGridPageChangedEventArgs**

*Example Syntax:*

**ToString**

```
[C#] public DataGridPageChangedEventArgs(object commandSource, int  
newPageIndex);
```

```
[C++] public: DataGridPageChangedEventArgs(Object* commandSource, int  
newPageIndex);
```

```
[VB] Public Sub New(ByVal commandSource As Object, ByVal newPageIndex  
As Integer)
```

```
[JScript] public function DataGridPageChangedEventArgs(commandSource :  
Object, newPageIndex : int);
```

### *Description*

Initializes a new instance of the **System.Web.UI.WebControls.DataGridPageChangedEventArgs** class.

Use this constructor to create and initialize a new instance of the **System.Web.UI.WebControls.DataGridPageChangedEventArgs** class. The source of the command. The index of the page selected by the user from the page selection element of the **System.Web.UI.WebControls.DataGrid** control.

**CommandSource**

**ToString**

1  
2 [C#] public object CommandSource {get;}

3 [C++] public: \_\_property Object\* get\_CommandSource();

4 [VB] Public ReadOnly Property CommandSource As Object

5 [JScript] public function get CommandSource() : Object;

6  
7 *Description*

8 Gets the source of the command.

9 Use the

10 **System.Web.UI.WebControls.DataGridPageChangedEventArgs.CommandS**  
11 **ource** property to determine the source of the command that raised the event.

12 Because this event is only raised when a button in the page selection element of  
13 the **System.Web.UI.WebControls.DataGrid** control is clicked, the returned  
14 source is a **System.Web.UI.WebControls.DataGridItem** that represents the page  
15 selection element.

16 NewPageIndex

17 ToString

18  
19 [C#] public int NewPageIndex {get;}

20 [C++] public: \_\_property int get\_NewPageIndex();

21 [VB] Public ReadOnly Property NewPageIndex As Integer

22 [JScript] public function get NewPageIndex() : int;

23  
24 *Description*

1 Gets the index of the page selected by the user in the page selection element  
2 of the **System.Web.UI.WebControls.DataGrid** control.

3 Use the  
4 **System.Web.UI.WebControls.DataGridPageChangedEventArgs.NewPageIndex**  
5 **ex** property to determine the index of the page selected by the user in the page  
6 selection element of the **System.Web.UI.WebControls.DataGrid** control. This  
7 value is often used to set the  
8 **System.Web.UI.WebControls.DataGrid.CurrentPageIndex** property of the  
9 **System.Web.UI.WebControls.DataGrid** control to display the selected page.

10 **DataGridPageChangedEventHandler** delegate  
11 (**System.Web.UI.WebControls**)  
12 **ToString**

### 15 *Description*

16 Represents the method that will handle the  
17 **System.Web.UI.WebControls.DataGrid.PageIndexChanged** event of the  
18 **System.Web.UI.WebControls.DataGrid** control. The source of the event. A  
19 **System.Web.UI.WebControls.DataGridPageChangedEventArgs** that contains  
20 the event data.

21 The **System.Web.UI.WebControls.DataGrid.PageIndexChanged** event  
22 is raised when a button in the page selection element of the  
23 **System.Web.UI.WebControls.DataGrid** control is clicked.

24 **DataGridPagerStyle** class (**System.Web.UI.WebControls**)  
25 **ToString**

1  
2  
3 *Description*

4 Specifies the style for the pager of the  
5 **System.Web.UI.WebControls.DataGrid** control. This class cannot be inherited.

6 The pager is an element on the **System.Web.UI.WebControls.DataGrid**  
7 control that allow you to link to other pages when paging is enabled. The  
8 **System.Web.UI.WebControls.DataGrid.PagerStyle** property of the  
9 **System.Web.UI.WebControls.DataGrid** uses an instance of this class to  
10 represent the style properties for the pager.

11 BackColor

12 BorderColor

13 BorderStyle

14 BorderWidth

15 Container

16 CssClass

17 DesignMode

18 Events

19 Font

20 ForeColor

21 Height

22 HorizontalAlign

23 IsEmpty

24 IsTrackingViewState

25 Mode

ToString

*Description*

Gets or sets a value that specifies whether the pager element displays buttons that link to the next and previous page, or numeric buttons that link directly to a page.

Use the **System.Web.UI.WebControls.DataGridPagerStyle.Mode** property to specify which set of pager buttons to use on the **System.Web.UI.WebControls.DataGrid** control. You can specify buttons that link to the next and previous page, or numeric buttons that link directly to a page.

NextPageText

ToString

[C#] public string NextPageText {get; set;}

[C++] public: \_\_property String\* get\_NextPageText();public: \_\_property void set\_NextPageText(String\*);

[VB] Public Property NextPageText As String

[JScript] public function get NextPageText() : String;public function set NextPageText(String);

*Description*

Gets or sets the text displayed for the next page button.

Use the

**System.Web.UI.WebControls.DataGridPagerStyle.NextPageText** property to



1 provide custom text for the next page button. The

2 **System.Web.UI.WebControls.DataGridPagerStyle.Mode** property must be set

3 to **PagerMode.NextPrev** for this property to have any effect.

4       PageButtonCount

5       ToString

6  
7 [C#] public int PageButtonCount {get; set;}

8 [C++] public: \_\_property int get\_PageButtonCount();public: \_\_property void

9 set\_PageButtonCount(int);

10 [VB] Public Property PageButtonCount As Integer

11 [JScript] public function get PageButtonCount() : int;public function set

12 PageButtonCount(int);

13  
14 *Description*

15       Gets or sets the number of numeric buttons to display concurrently in the  
16 pager element of the **System.Web.UI.WebControls.DataGrid** control.

17       Use the

18 **System.Web.UI.WebControls.DataGridPagerStyle.PageButtonCount** property

19 to specify the number of numeric buttons to display concurrently in the pager

20 element of the **System.Web.UI.WebControls.DataGrid** control. The

21 **System.Web.UI.WebControls.DataGridPagerStyle.Mode** property must be set

22 to **PagerMode.NumericPages** for this property to have any effect.

23       Position

24       ToString

1  
2 [C#] public PagerPosition Position {get; set;}

3 [C++] public: \_\_property PagerPosition get \_Position();public: \_\_property void  
4 set \_Position(PagerPosition);

5 [VB] Public Property Position As PagerPosition

6 [JScript] public function get Position() : PagerPosition;public function set  
7 Position(PagerPosition);

8  
9 *Description*

10 Gets or sets the position of the pager element in the  
11 **System.Web.UI.WebControls.DataGrid** control.

12 Use the **System.Web.UI.WebControls.DataGridPagerStyle.Position**  
13 property to specify the location where the pager element is displayed in the  
14 **System.Web.UI.WebControls.DataGrid** control. The pager element can be  
15 displayed at the upper edge, the lower edge, or at both the upper and lower edges  
16 of the **System.Web.UI.WebControls.DataGrid** control.

17 PrevPageText

18 ToString

19  
20 [C#] public string PrevPageText {get; set;}

21 [C++] public: \_\_property String\* get \_PrevPageText();public: \_\_property void  
22 set \_PrevPageText(String\*);

23 [VB] Public Property PrevPageText As String

24 [JScript] public function get PrevPageText() : String;public function set  
25 PrevPageText(String);

1  
2 *Description*

3 Gets or sets the text displayed for the previous page button.

4 Use the

5 **System.Web.UI.WebControls.DataGridPagerStyle.PrevPageText** property to  
6 provide custom text for the next page button. The

7 **System.Web.UI.WebControls.DataGridPagerStyle.Mode** property must be set  
8 to **PagerMode.NextPrev** for this property to have any effect.

9 Site

10 VerticalAlign

11 ViewState

12 Visible

13 ToString

14  
15  
16 *Description*

17 Gets or sets a value indicating whether the pager is displayed in the  
18 **System.Web.UI.WebControls.DataGrid** control.

19 Use the **System.Web.UI.WebControls.DataGridPagerStyle.Visible**  
20 property to specify whether the pager is displayed in the

21 **System.Web.UI.WebControls.DataGrid** control.

22 Width

23 Wrap

24 CopyFrom  
25

```

1
2 [C#] public override void CopyFrom(Style s);
3 [C++] public: void CopyFrom(Style* s);
4 [VB] Overrides Public Sub CopyFrom(ByVal s As Style)
5 [JScript] public override function CopyFrom(s : Style);
6

```

### *Description*

Copies the style of the specified **System.Web.UI.WebControls.Style** object into this instance of the **System.Web.UI.WebControls.DataGridPagerStyle** class.

Use the **System.Web.UI.WebControls.DataGridPagerStyle.CopyFrom(System.Web.UI.WebControls.Style)** method to copy the style properties of the specified **System.Web.UI.WebControls.Style** object into this instance of the **System.Web.UI.WebControls.DataGridPagerStyle** class. The **System.Web.UI.WebControls.Style** object to copy from.

### *MergeWith*

```

18
19 [C#] public override void MergeWith(Style s);
20 [C++] public: void MergeWith(Style* s);
21 [VB] Overrides Public Sub MergeWith(ByVal s As Style)
22 [JScript] public override function MergeWith(s : Style);
23

```

### *Description*

Merges the style of the specified **System.Web.UI.WebControls.Style** object with this instance of the **System.Web.UI.WebControls.DataGridPagerStyle** class.

Use the **System.Web.UI.WebControls.DataGridPagerStyle.MergeWith(System.Web.UI.WebControls.Style)** method to combine the style properties of the specified **System.Web.UI.WebControls.Style** object with this instance of the **System.Web.UI.WebControls.DataGridPagerStyle** class. If a property from this instance is already set to a value, the property is unchanged. If a property is not set, this method sets that property with the value from the corresponding property of the **System.Web.UI.WebControls.Style** object. The **System.Web.UI.WebControls.Style** object to merge with.

#### Reset

[C#] public override void Reset();  
[C++] public: void Reset();  
[VB] Overrides Public Sub Reset()  
[JScript] public override function Reset();

#### *Description*

Restores the **System.Web.UI.WebControls.DataGridPagerStyle** object to its default values.

Use the **System.Web.UI.WebControls.DataGridPagerStyle.Reset** method to restore the **System.Web.UI.WebControls.DataGridPagerStyle** object to its default values.

1       DataGridSortCommandEventArgs class (System.Web.UI.WebControls)

2       TrackViewState

3  
4  
5       *Description*

6       Provides data for the

7       **System.Web.UI.WebControls.DataGrid.SortCommand** event of the

8       **System.Web.UI.WebControls.DataGrid** control. This class cannot be inherited.

9       When sorting is enabled by setting the

10      **System.Web.UI.WebControls.DataGrid.AllowSorting** property of the

11      **System.Web.UI.WebControls.DataGrid** control to **true** ,

12      **System.Web.UI.WebControls.LinkButton** controls are rendered in the header of  
13      each column that has the

14      **System.Web.UI.WebControls.DataGridColumn.SortExpression** property set.

15      (For automatically generated columns, the

16      **System.Web.UI.WebControls.DataGridColumn.SortExpression** property

17      contains the same value as the data field.) These links allow you to sort the

18      **System.Web.UI.WebControls.DataGrid** control by the selected column. The

19      **System.Web.UI.WebControls.DataGrid.SortCommand** event is raised when a

20      **System.Web.UI.WebControls.LinkButton** control is clicked.

21       DataGridSortCommandEventArgs

22       *Example Syntax:*

23       TrackViewState

24  
25      [C#] public DataGridSortCommandEventArgs(object commandSource,

1 DataGridCommandEventArgs dce);

2 [C++] public: DataGridSortCommandEventArgs(Object\* commandSource,

3 DataGridCommandEventArgs\* dce);

4 [VB] Public Sub New(ByVal commandSource As Object, ByVal dce As

5 DataGridCommandEventArgs)

6 [JScript] public function DataGridSortCommandEventArgs(commandSource :

7 Object, dce : DataGridCommandEventArgs);

9 *Description*

10       Initializes a new instance of the  
11 **System.Web.UI.WebControls.DataGridSortCommandEventArgs** class.

12       Use this constructor to create and initialize a new instance of the  
13 **System.Web.UI.WebControls.DataGridSortCommandEventArgs** class. The  
14 source of the command. A  
15 **System.Web.UI.WebControls.DataGridCommandEventArgs** that contains the  
16 event data.

17       CommandSource

18       TrackViewState

20 [C#] public object CommandSource {get;}

21 [C++] public: \_\_property Object\* get\_CommandSource();

22 [VB] Public ReadOnly Property CommandSource As Object

23 [JScript] public function get CommandSource() : Object;

25 *Description*

Gets the source of the command.

Use the

**System.Web.UI.WebControls.DataGridSortCommandEventArgs.CommandSource** property to get the command source that raises the **System.Web.UI.WebControls.DataGrid.SortCommand** event. Because this event is only raised when a link is clicked in the header of the **System.Web.UI.WebControls.DataGrid** control, the returned source is a **System.Web.UI.WebControls.DataGridItem** that represents the header. This property can be used to programmatically control the header.

**SortExpression**

**TrackViewState**

[C#] public string SortExpression {get;}

[C++] public: \_\_property String\* get\_SortExpression();

[VB] Public ReadOnly Property SortExpression As String

[JScript] public function get SortExpression() : String;

### *Description*

Gets the expression used to sort the

**System.Web.UI.WebControls.DataGrid** control.

Use the

**System.Web.UI.WebControls.DataGridSortCommandEventArgs.SortExpression** property to determine which column the user selects to sort the **System.Web.UI.WebControls.DataGrid** control.



1       DataGridSortCommandEventHandler delegate  
2       (System.Web.UI.WebControls)

3       ToString

4  
5  
6       *Description*

7       Represents the method that will handle the  
8       **System.Web.UI.WebControls.DataGrid.SortCommand** event of the  
9       **System.Web.UI.WebControls.DataGrid** control. The source of the event. A  
10      **System.Web.UI.WebControls.DataGridSortCommandEventArgs** that contains  
11      the event data.

12      When sorting is enabled by setting the  
13      **System.Web.UI.WebControls.DataGrid.AllowSorting** property of the  
14      **System.Web.UI.WebControls.DataGrid** control to **true** ,  
15      **System.Web.UI.WebControls.LinkButton** controls are rendered in the header of  
16      each column. These links allow you to sort the  
17      **System.Web.UI.WebControls.DataGrid** control by the selected column. The  
18      **System.Web.UI.WebControls.DataGrid.SortCommand** event is raised when a  
19      **System.Web.UI.WebControls.LinkButton** control is clicked.

20      DataKeyCollection class (System.Web.UI.WebControls)

21      ToString

22  
23  
24      *Description*  
25

Represents a collection of primary key field names. This class cannot be inherited.

**DataKeyCollection**

*Example Syntax:*

**ToString**

[C#] public DataKeyCollection(ArrayList keys);

[C++] public: DataKeyCollection(ArrayList\* keys);

[VB] Public Sub New(ByVal keys As ArrayList)

[JScript] public function DataKeyCollection(keys : ArrayList);

#### *Description*

Initializes a new instance of the **System.Web.UI.WebControls.DataKeyCollection** class. A **System.Collections.ArrayList** to store the keys.

**Count**

**ToString**

[C#] public int Count {get;}

[C++] public: \_\_property int get\_Count();

[VB] Public ReadOnly Property Count As Integer

[JScript] public function get Count() : int;

#### *Description*

Gets the number of objects in the collection. This property is read-only.

1       IsReadOnly  
2       ToString  
3  
4   [C#] public bool IsReadOnly {get;}  
5   [C++] public: \_\_property bool get\_IsReadOnly();  
6   [VB] Public ReadOnly Property IsReadOnly As Boolean  
7   [JScript] public function get IsReadOnly() : Boolean;

8  
9   *Description*

10       Gets the value that specifies whether items in the  
11   **System.Web.UI.WebControls.DataKeyCollection** can be modified. This  
12   property is read-only.

13       IsSynchronized  
14       ToString  
15  
16   [C#] public bool IsSynchronized {get;}  
17   [C++] public: \_\_property bool get\_IsSynchronized();  
18   [VB] Public ReadOnly Property IsSynchronized As Boolean  
19   [JScript] public function get IsSynchronized() : Boolean;

20  
21   *Description*

22       Gets a value that indicates whether the  
23   **System.Web.UI.WebControls.DataKeyCollection** is thread-safe. This property  
24   is read-only.

25       Item

## ToString

[C#] public object this[int index] {get;}

[C++] public: \_\_property Object\* get\_Item(int index);

[VB] Public Default ReadOnly Property Item(ByVal index As Integer) As Object

[JScript] returnValue = DataKeyCollectionObject.Item(index);

### *Description*

Gets the primary key field name at the specified index in the collection. This property is read-only. The index of the primary key field name in the collection to retrieve.

## SyncRoot

## ToString

[C#] public object SyncRoot {get;}

[C++] public: \_\_property Object\* get\_SyncRoot();

[VB] Public ReadOnly Property SyncRoot As Object

[JScript] public function get SyncRoot() : Object;

### *Description*

Gets the object used to synchronize access to the collection. This property is read-only.

## CopyTo

[C#] public void CopyTo(Array array, int index);

1 [C++] public: \_\_sealed void CopyTo(Array\* array, int index);

2 [VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As  
3 Integer)

4 [JScript] public function CopyTo(array : Array, index : int);

5  
6 *Description*

7 Copies the contents of the entire collection into an **System.Array**  
8 appending at the specified index of the **System.Array** . The **System.Array** to  
9 copy the contents of the collection into. The index of the **System.Array** to begin  
10 copying the contents of the collection into.

11 GetEnumerator

12  
13 [C#] public IEnumerator GetEnumerator();

14 [C++] public: \_\_sealed IEnumerator\* GetEnumerator();

15 [VB] NotOverridable Public Function GetEnumerator() As IEnumerator

16 [JScript] public function GetEnumerator() : IEnumerator;

17  
18 *Description*

19 Creates an enumerator for the  
20 **System.Web.UI.WebControls.DataKeyCollection** used to iterate through the  
21 collection.

22 DataList class (System.Web.UI.WebControls)

23 ToString

1  
2  
3 *Description*

4 A data bound list control that displays items using templates.

5 Use the **System.Web.UI.WebControls.DataList** control to display a  
6 template-defined data bound list. The **System.Web.UI.WebControls.DataList**  
7 control supports selecting and editing.

8 ToString

9  
10 [C#] public const string CancelCommandName;

11 [C++] public: const String\* CancelCommandName;

12 [VB] Public Const CancelCommandName As String

13 [JScript] public var CancelCommandName : String;

14  
15 *Description*

16 Represents the **Cancel** command name. This field is read-only.

17 Use the **System.Web.UI.WebControls.DataList.CancelCommandName**  
18 field to represent the **Cancel** command name.

19 ToString

20  
21 [C#] public const string DeleteCommandName;

22 [C++] public: const String\* DeleteCommandName;

23 [VB] Public Const DeleteCommandName As String

24 [JScript] public var DeleteCommandName : String;

1  
2 *Description*

3 Represents the **Delete** command name. This field is read-only.

4 Use the **System.Web.UI.WebControls.DataList.DeleteCommandName**  
5 field to represent the **Delete** command name.

6 ToString

7  
8 [C#] public const string EditCommandName;

9 [C++] public: const String\* EditCommandName;

10 [VB] Public Const EditCommandName As String

11 [JScript] public var EditCommandName : String;

12  
13 *Description*

14 Represents the **Edit** command name. This field is read-only.

15 Use the **System.Web.UI.WebControls.DataList.EditCommandName**  
16 field to represent the **Edit** command name.

17 ToString

18  
19 [C#] public const string SelectCommandName;

20 [C++] public: const String\* SelectCommandName;

21 [VB] Public Const SelectCommandName As String

22 [JScript] public var SelectCommandName : String;

23  
24 *Description*

25 Represents the **Select** command name. This field is read-only.

1        Use the **System.Web.UI.WebControls.DataList.SelectCommandName**  
2 field to represent the **Select** command name.

3        ToString

4  
5        [C#] public const string UpdateCommandName;  
6        [C++] public: const String\* UpdateCommandName;  
7        [VB] Public Const UpdateCommandName As String  
8        [JScript] public var UpdateCommandName : String;

9  
10        *Description*

11        Represents the **Update** command name. This field is read-only.

12        Use the **System.Web.UI.WebControls.DataList.UpdateCommandName**  
13 field to represent the **Update** command name.

14        DataList

15        *Example Syntax:*

16        ToString

17  
18        [C#] public DataList();  
19        [C++] public: DataList();  
20        [VB] Public Sub New()  
21        [JScript] public function DataList();

22  
23        *Description*

24        Initializes a new instance of the **System.Web.UI.WebControls.DataList**  
25 class.



1 Use this constructor to create and initialize a new instance of the  
2 **System.Web.UI.WebControls.DataList** class.

3 AccessKey

4 AlternatingItemStyle

5 ToString

6  
7  
8 *Description*

9 Gets the style properties for alternating items in the  
10 **System.Web.UI.WebControls.DataList** control.

11 Use the **System.Web.UI.WebControls.DataList.AlternatingItemStyle**  
12 property to provide a custom style for the alternating items in the  
13 **System.Web.UI.WebControls.DataList** control. Common style attributes that  
14 can be adjusted include forecolor, backcolor, font, and content alignment within  
15 the cell. Providing a different style enhances the appearance of the  
16 **System.Web.UI.WebControls.DataList** control.

17 AlternatingItemTemplate

18 ToString

19  
20 [C#] public virtual ITemplate AlternatingItemTemplate {get; set;}

21 [C++] public: \_\_property virtual ITemplate\*

22 get\_AlternatingItemTemplate();public: \_\_property virtual void

23 set\_AlternatingItemTemplate(ITemplate\*);

24 [VB] Overridable Public Property AlternatingItemTemplate As ITemplate

25 [JScript] public function get AlternatingItemTemplate() : ITemplate;public

1 function set AlternatingItemTemplate(ITemplate);

2  
3 *Description*

4 Gets or sets the template for alternating items in the  
5 **System.Web.UI.WebControls.DataList** .

6 Use the  
7 **System.Web.UI.WebControls.DataList.AlternatingItemTemplate** property to  
8 control the contents of alternating items in the  
9 **System.Web.UI.WebControls.DataList** control. The appearance of alternating  
10 items is controlled by the  
11 **System.Web.UI.WebControls.DataList.AlternatingItemStyle** property.

12 Attributes

13 BackColor

14 BorderColor

15 BorderStyle

16 BorderWidth

17 CellPadding

18 CellSpacing

19 ChildControlsCreated

20 ClientID

21 Context

22 Controls

23 ControlStyle

24 ControlStyleCreated

25 CssClass

1 DataKeyField  
2 DataKeys  
3 DataKeysArray  
4 DataMember  
5 DataSource  
6 EditItemIndex  
7 ToString

8  
9  
10 *Description*

11 Gets or sets the index number of the selected item in the  
12 **System.Web.UI.WebControls.DataList** control to edit.

13 Use the **System.Web.UI.WebControls.DataList.EditItemIndex** property  
14 to programmatically specify the item selected for editing. You can also use this  
15 property to determine the index of the item selected for editing.

16 EditItemStyle

17 ToString

18  
19 [C#] public virtual TableItemStyle EditItemStyle {get;}

20 [C++] public: \_\_property virtual TableItemStyle\* get\_EditItemStyle();

21 [VB] Overridable Public ReadOnly Property EditItemStyle As TableItemStyle

22 [JScript] public function get EditItemStyle() : TableItemStyle;

23  
24 *Description*  
25

1 Gets the style properties for the item selected for editing in the  
2 **System.Web.UI.WebControls.DataList** control.

3 Use the **System.Web.UI.WebControls.DataList.EditItemStyle** property  
4 to provide a custom style for the item selected for editing in the  
5 **System.Web.UI.WebControls.DataList** control. Common style attributes that  
6 can be adjusted include forecolor, backcolor, font, and content alignment within  
7 the cell. Providing a different style enhances the appearance of the  
8 **System.Web.UI.WebControls.DataList** control.

9 EditItemTemplate

10 ToString

11  
12 [C#] public virtual ITemplate EditItemTemplate {get; set;}

13 [C++] public: \_\_property virtual ITemplate\* get\_EditItemTemplate();public:

14 \_\_property virtual void set\_EditItemTemplate(ITemplate\*);

15 [VB] Overridable Public Property EditItemTemplate As ITemplate

16 [JScript] public function get EditItemTemplate() : ITemplate;public function set  
17 EditItemTemplate(ITemplate);

18  
19 *Description*

20 Gets or sets the template for the item selected for editing in the  
21 **System.Web.UI.WebControls.DataList** control.

22 Use the **System.Web.UI.WebControls.DataList.EditItemTemplate**  
23 property to control the contents of the item selected for editing in the  
24 **System.Web.UI.WebControls.DataList** control. The appearance of the item  
25

selected for editing is controlled by the

**System.Web.UI.WebControls.DataList.EditItemStyle** property.

Enabled

EnableViewState

Events

ExtractTemplateRows

ToString

#### *Description*

Gets or sets a value that indicates whether the rows of a **System.Web.UI.WebControls.Table** control, defined in each template of a **System.Web.UI.WebControls.DataList** control, are extracted and displayed.

The contents of the **System.Web.UI.WebControls.DataList** control are specified by using templates. Normally, you list controls that you want to display in the templates. You can also place a **System.Web.UI.WebControls.Table** control in a template and display the rows of the table.

Font

FooterStyle

ToString

#### *Description*

Gets the style properties for the footer section of the **System.Web.UI.WebControls.DataList** control.

Use this property to provide a custom style for the footer section of the **System.Web.UI.WebControls.DataList** control. Common style attributes that can be adjusted include forecolor, backcolor, font, and content alignment within the cell. Providing a different style enhances the appearance of the **System.Web.UI.WebControls.DataList** control.

FooterTemplate

ToString

[C#] public virtual ITemplate FooterTemplate {get; set;}

[C++] public: \_\_property virtual ITemplate\* get\_FooterTemplate();public:

\_\_property virtual void set\_FooterTemplate(ITemplate\*);

[VB] Overridable Public Property FooterTemplate As ITemplate

[JScript] public function get FooterTemplate() : ITemplate;public function set

FooterTemplate(ITemplate);

#### *Description*

Gets or sets the template for the footer section of the **System.Web.UI.WebControls.DataList** control.

Use the **System.Web.UI.WebControls.DataList.FooterTemplate** property to control the contents of the footer section. The appearance of the footer section is controlled by the **System.Web.UI.WebControls.DataList.FooterStyle** property.

ForeColor

GridLines

ToString

1  
2  
3 *Description*

4 Gets or sets the grid line style for the  
5 **System.Web.UI.WebControls.DataList** control when the  
6 **System.Web.UI.WebControls.DataList.RepeatLayout** property is set to  
7 **RepeatLayout.Table** .

8 Use the **System.Web.UI.WebControls.DataList.GridLines** property to  
9 specify the grid line style for the **System.Web.UI.WebControls.DataList** control.

10 The following table lists the possible styles.

11 HasChildViewState

12 HeaderStyle

13 ToString

14  
15  
16 *Description*

17 Gets the style properties for the heading section of the  
18 **System.Web.UI.WebControls.DataList** control.

19 Use this property to provide a custom style for the heading of the  
20 **System.Web.UI.WebControls.DataList** control. Common style attributes that  
21 can be adjusted include forecolor, backcolor, font, and content alignment within  
22 the cell. Providing a different style enhances the appearance of the  
23 **System.Web.UI.WebControls.DataList** control.

24 HeaderTemplate

25 ToString

```

1
2 [C#] public virtual ITemplate HeaderTemplate {get; set;}
3 [C++] public: __property virtual ITemplate* get_HeaderTemplate();public:
4 __property virtual void set_HeaderTemplate(ITemplate*);
5 [VB] Overridable Public Property HeaderTemplate As ITemplate
6 [JScript] public function get HeaderTemplate() : ITemplate;public function set
7 HeaderTemplate(ITemplate);
8

```

### *Description*

Gets or sets the template for the heading section of the **System.Web.UI.WebControls.DataList** control.

Use the **System.Web.UI.WebControls.DataList.HeaderTemplate** property to control the contents of the heading section. The appearance of the header section is controlled by the **System.Web.UI.WebControls.DataList.HeaderStyle** property.

Height

HorizontalAlign

ID

IsTrackingViewState

Items

ToString

### *Description*



1 Gets a collection of **System.Web.UI.WebControls.DataListItem** objects  
2 representing the individual items within the control.

3 Use the **System.Web.UI.WebControls.DataList.Items** collection to  
4 programmatically control the items in the  
5 **System.Web.UI.WebControls.DataList** control. The  
6 **System.Web.UI.WebControls.DataList.Items** collection does not provide any  
7 methods to add or remove items to the collection. However, you can control the  
8 contents of an item by providing a handler for the  
9 **System.Web.UI.WebControls.DataList.ItemCreated** event.

10 **ItemStyle**

11 **ToString**

12  
13 [C#] public virtual TableItemStyle ItemStyle {get;}

14 [C++] public: \_\_property virtual TableItemStyle\* get\_ItemStyle();

15 [VB] Overridable Public ReadOnly Property ItemStyle As TableItemStyle

16 [JScript] public function get ItemStyle() : TableItemStyle;

17  
18 *Description*

19 Gets the style properties for the items in the  
20 **System.Web.UI.WebControls.DataList** control.

21 Use this property to provide a custom style for the items of the  
22 **System.Web.UI.WebControls.DataList** control. Common style attributes that  
23 can be adjusted include forecolor, backcolor, font, and content alignment within  
24 the cell. Providing a different style enhances the appearance of the  
25 **System.Web.UI.WebControls.DataList** control.

1       ItemTemplate

2       ToString

3

4   [C#] public virtual ITemplate ItemTemplate {get; set;}

5   [C++] public: \_\_property virtual ITemplate\* get\_ItemTemplate();public:

6   \_\_property virtual void set\_ItemTemplate(ITemplate\*);

7   [VB] Overridable Public Property ItemTemplate As ITemplate

8   [JScript] public function get ItemTemplate() : ITemplate;public function set

9   ItemTemplate(ITemplate);

10

11   *Description*

12       Gets or sets the template for the items in the

13   **System.Web.UI.WebControls.DataList** control.

14       Use the **System.Web.UI.WebControls.DataList.ItemTemplate** property

15   to control the contents of the items in the

16   **System.Web.UI.WebControls.DataList** control. The appearance of the items in

17   the **System.Web.UI.WebControls.DataList** control is controlled by the

18   **System.Web.UI.WebControls.DataList.ItemStyle** property.

19       NamingContainer

20       Page

21       Parent

22       RepeatColumns

23       ToString

24

25

1  
2  
3 *Description*

4 Gets or sets the number of columns to display in the  
5 **System.Web.UI.WebControls.DataList** control.

6 Use this property to specify the number of columns that display items in the  
7 **System.Web.UI.WebControls.DataList** control. For example, if you set this  
8 property to 5 , the **System.Web.UI.WebControls.DataList** control displays its  
9 items in five columns.

10 RepeatDirection

11 ToString

12  
13 [C#] public virtual RepeatDirection RepeatDirection {get; set;}

14 [C++] public: \_\_property virtual RepeatDirection get\_RepeatDirection();public:  
15 \_\_property virtual void set\_RepeatDirection(RepeatDirection);

16 [VB] Overridable Public Property RepeatDirection As RepeatDirection

17 [JScript] public function get RepeatDirection() : RepeatDirection;public function  
18 set RepeatDirection(RepeatDirection);

19  
20 *Description*

21 Gets or sets whether the **System.Web.UI.WebControls.DataList** control  
22 displays vertically or horizontally.

23 Use the **System.Web.UI.WebControls.DataList.RepeatDirection**  
24 property to specify the display direction of the  
25 **System.Web.UI.WebControls.DataList** control.

RepeatLayout

ToString

[C#] public virtual RepeatLayout RepeatLayout {get; set;}

[C++] public: \_\_property virtual RepeatLayout get\_RepeatLayout();public:

\_\_property virtual void set\_RepeatLayout(RepeatLayout);

[VB] Overridable Public Property RepeatLayout As RepeatLayout

[JScript] public function get RepeatLayout() : RepeatLayout;public function set

RepeatLayout(RepeatLayout);

### *Description*

Gets or sets whether the control is displayed in a table or flow layout.

Use the **System.Web.UI.WebControls.DataList.RepeatLayout** property to specify whether the items in the **System.Web.UI.WebControls.DataList** control are displayed in a table. If this property is set to **RepeatLayout.Table**, the items in the list are displayed in a table. If this property is set to **RepeatLayout.Flow**, the items in the list are displayed without a table structure.

SelectedIndex

ToString

[C#] public virtual int SelectedIndex {get; set;}

[C++] public: \_\_property virtual int get\_SelectedIndex();public: \_\_property virtual

void set\_SelectedIndex(int);

[VB] Overridable Public Property SelectedIndex As Integer

[JScript] public function get SelectedIndex() : int;public function set

1 SelectedIndex(int);

3 *Description*

4 Gets or sets the index of the selected item in the  
5 **System.Web.UI.WebControls.DataList** control.

6 Use the **System.Web.UI.WebControls.DataList.SelectedIndex** property  
7 to programmatically specify the selected item in the  
8 **System.Web.UI.WebControls.DataList** control. You can also use this property  
9 to determine the index of the selected item.

10 SelectedItem

11 ToString

13 [C#] public virtual DataListItem SelectedItem {get;}

14 [C++] public: \_\_property virtual DataListItem\* get\_SelectedItem();

15 [VB] Overridable Public ReadOnly Property SelectedItem As DataListItem

16 [JScript] public function get SelectedItem() : DataListItem;

18 *Description*

19 Gets the selected item in the **System.Web.UI.WebControls.DataList**  
20 control.

21 Use the **System.Web.UI.WebControls.DataList.SelectedItem** property to  
22 get a **System.Web.UI.WebControls.DataListItem** object that represents the  
23 selected item in the **System.Web.UI.WebControls.DataList** control. This object  
24 can then be used to access the properties of the selected item.

25 SelectedItemStyle

ToString

```
[C#] public virtual TableItemStyle SelectedItemStyle {get;}
[C++] public: __property virtual TableItemStyle* get_SelectedItemStyle();
[VB] Overridable Public ReadOnly Property SelectedItemStyle As TableItemStyle
[JScript] public function get SelectedItemStyle() : TableItemStyle;
```

### *Description*

Gets the style properties for the selected item in the **System.Web.UI.WebControls.DataList** control.

Use this property to provide a custom style for the selected item in the **System.Web.UI.WebControls.DataList** control. Common style attributes that can be adjusted include forecolor, backcolor, font, and content alignment within the cell. Providing a different style enhances the appearance of the **System.Web.UI.WebControls.DataList** control.

SelectedItemTemplate

ToString

```
[C#] public virtual ITemplate SelectedItemTemplate {get; set;}
[C++] public: __property virtual ITemplate* get_SelectedItemTemplate();public:
__property virtual void set_SelectedItemTemplate(ITemplate*);
[VB] Overridable Public Property SelectedItemTemplate As ITemplate
[JScript] public function get SelectedItemTemplate() : ITemplate;public function
set SelectedItemTemplate(ITemplate);
```

1  
2 *Description*

3 Gets or sets the template for the selected item in the  
4 **System.Web.UI.WebControls.DataList** control.

5 Use the **System.Web.UI.WebControls.DataList.SelectedItemTemplate**  
6 property to control the contents of the selected item. The appearance of the  
7 selected item is controlled by the  
8 **System.Web.UI.WebControls.DataList.SelectedItemStyle** property.

9 SeparatorStyle

10 ToString

11  
12 [C#] public virtual TableItemStyle SeparatorStyle {get;}

13 [C++] public: \_\_property virtual TableItemStyle\* get\_SeparatorStyle();

14 [VB] Overridable Public ReadOnly Property SeparatorStyle As TableItemStyle

15 [JScript] public function get SeparatorStyle() : TableItemStyle;

16  
17 *Description*

18 Gets the style properties of the separator between each item in the  
19 **System.Web.UI.WebControls.DataList** control.

20 Use the **System.Web.UI.WebControls.DataList.SeparatorStyle** property  
21 to provide a custom style for the separator between each item in the  
22 **System.Web.UI.WebControls.DataList** control. The separator allows you to  
23 place an element with custom content between each item in the  
24 **System.Web.UI.WebControls.DataList** control. Common style attributes that  
25 can be adjusted include forecolor, bgcolor, font, and content alignment within

the cell. Providing a different style enhances the appearance of the

**System.Web.UI.WebControls.DataList** control.

SeparatorTemplate

ToString

[C#] public virtual ITemplate SeparatorTemplate {get; set;}

[C++] public: \_\_property virtual ITemplate\* get\_SeparatorTemplate();public:

\_\_property virtual void set\_SeparatorTemplate(ITemplate\*);

[VB] Overridable Public Property SeparatorTemplate As ITemplate

[JScript] public function get SeparatorTemplate() : ITemplate;public function set

SeparatorTemplate(ITemplate);

### *Description*

Gets or sets the template for the separator between the items of the

**System.Web.UI.WebControls.DataList** control.

Use the **System.Web.UI.WebControls.DataList.SeparatorTemplate**

property to control the contents of the separator between the items of the

**System.Web.UI.WebControls.DataList** control. The separator allows you to

place an element with custom content between each item in the

**System.Web.UI.WebControls.DataList** control. The appearance of the separator

between the items of the **System.Web.UI.WebControls.DataList** control is

controlled by the **System.Web.UI.WebControls.DataList.SeparatorStyle**

property.

ShowFooter

ToString



```

1
2 [C#] public virtual bool ShowFooter {get; set;}
3 [C++] public: __property virtual bool get_ShowFooter();public: __property virtual
4 void set_ShowFooter(bool);
5 [VB] Overridable Public Property ShowFooter As Boolean
6 [JScript] public function get ShowFooter() : Boolean;public function set
7 ShowFooter(Boolean);
8

```

#### *Description*

Gets or sets a value indicating whether the footer section is displayed in the **System.Web.UI.WebControls.DataList** control.

Use the **System.Web.UI.WebControls.DataList.ShowFooter** property to specify whether the footer section is displayed in the **System.Web.UI.WebControls.DataList** control.

ShowHeader

ToString

```

15
16
17
18 [C#] public virtual bool ShowHeader {get; set;}
19 [C++] public: __property virtual bool get_ShowHeader();public: __property
20 virtual void set_ShowHeader(bool);
21 [VB] Overridable Public Property ShowHeader As Boolean
22 [JScript] public function get ShowHeader() : Boolean;public function set
23 ShowHeader(Boolean);
24

```

#### *Description*

1 Gets or sets a value indicating whether the header section is displayed in  
2 the **System.Web.UI.WebControls.DataList** control.

3 Use the **System.Web.UI.WebControls.DataList.ShowHeader** property to  
4 specify whether the header section is displayed in the  
5 **System.Web.UI.WebControls.DataList** control.

6 Site

7 Style

8 TabIndex

9 TagKey

10 TagName

11 TemplateSourceDirectory

12 ToolTip

13 UniqueID

14 ViewState

15 ViewStateIgnoresCase

16 Visible

17 Width

18 ToString

19  
20  
21 *Description*

22 Occurs when the **Cancel** button is clicked for an item in the  
23 **System.Web.UI.WebControls.DataList** control.  
24  
25

1       The **System.Web.UI.WebControls.DataList.CancelCommand** event is  
2 raised when the **Cancel** button is clicked for an item in the  
3 **System.Web.UI.WebControls.DataList** control.

4       ToString

5  
6  
7       *Description*

8       Occurs when the **Delete** button is clicked for an item in the  
9 **System.Web.UI.WebControls.DataList** control.

10       The **System.Web.UI.WebControls.DataList.DeleteCommand** event is  
11 raised when the **Delete** button is clicked for an item in the  
12 **System.Web.UI.WebControls.DataList** control.

13       ToString

14  
15  
16       *Description*

17       Occurs when the **Edit** button is clicked for an item in the  
18 **System.Web.UI.WebControls.DataList** control.

19       The **System.Web.UI.WebControls.DataList.EditCommand** event is  
20 raised when the **Edit** button is clicked for an item in the  
21 **System.Web.UI.WebControls.DataList** control.

22       ToString

23  
24  
25       *Description*

Occurs when any button is clicked in the  
**System.Web.UI.WebControls.DataList** control.

The **System.Web.UI.WebControls.DataList.ItemCommand** event is raised when any button is clicked in the **System.Web.UI.WebControls.DataList** control and is commonly used when you have a button control with a custom **CommandName** value.

ToString

[C#] public event DataListItemEventHandler ItemCreated;  
[C++] public: \_\_event DataListItemEventHandler\* ItemCreated;  
[VB] Public Event ItemCreated As DataListItemEventHandler

#### *Description*

Occurs on the server when an item in the  
**System.Web.UI.WebControls.DataList** control is created.

The **System.Web.UI.WebControls.DataList.ItemCreated** event is raised when an item in the **System.Web.UI.WebControls.DataList** control is created, both during round-trips and at data bind time.

ToString

[C#] public event DataListItemEventHandler ItemDataBound;  
[C++] public: \_\_event DataListItemEventHandler\* ItemDataBound;  
[VB] Public Event ItemDataBound As DataListItemEventHandler

#### *Description*

Occurs when an item is data bound to the  
**System.Web.UI.WebControls.DataList** control.

The **System.Web.UI.WebControls.DataList.ItemDataBound** event is raised after an item is data bound to the **System.Web.UI.WebControls.DataList** control. This event provides you with the last opportunity to access the data item before it is displayed on the client. After this event is raised, the data item is nulled out and no longer available.

**ToString**

#### *Description*

Occurs when the **Update** button is clicked for an item in the **System.Web.UI.WebControls.DataList** control.

The **System.Web.UI.WebControls.DataList.UpdateCommand** event is raised when the **Update** button for an item is clicked.

**CreateControlHierarchy**

[C#] protected override void CreateControlHierarchy(bool useDataSource);

[C++] protected: void CreateControlHierarchy(bool useDataSource);

[VB] Overrides Protected Sub CreateControlHierarchy(ByVal useDataSource As Boolean)

[JScript] protected override function CreateControlHierarchy(useDataSource : Boolean);

#### *Description*

## CreateControlStyle

[C#] protected override Style CreateControlStyle();

[C++] protected: Style\* CreateControlStyle();

[VB] Overrides Protected Function CreateControlStyle() As Style

[JScript] protected override function CreateControlStyle() : Style;

### *Description*

## CreateItem

[C#] protected virtual DataListItem CreateItem(int itemIndex, ListItemType itemType);

[C++] protected: virtual DataListItem\* CreateItem(int itemIndex, ListItemType itemType);

[VB] Overridable Protected Function CreateItem(ByVal itemIndex As Integer, ByVal itemType As ListItemType) As DataListItem

[JScript] protected function CreateItem(itemIndex : int, itemType : ListItemType) : DataListItem;

### *Description*

## InitializeItem

[C#] protected virtual void InitializeItem(DataListItem item);

[C++] protected: virtual void InitializeItem(DataListItem\* item);

1 [VB] Overridable Protected Sub InitializeItem(ByVal item As DataListItem)

2 [JScript] protected function InitializeItem(item : DataListItem);

3 LoadViewState

5 [C#] protected override void LoadViewState(object savedState);

6 [C++] protected: void LoadViewState(Object\* savedState);

7 [VB] Overrides Protected Sub LoadViewState(ByVal savedState As Object)

8 [JScript] protected override function LoadViewState(savedState : Object);

10 *Description*

11 OnBubbleEvent

13 [C#] protected override bool OnBubbleEvent(object source, EventArgs e);

14 [C++] protected: bool OnBubbleEvent(Object\* source, EventArgs\* e);

15 [VB] Overrides Protected Function OnBubbleEvent(ByVal source As Object,

16 ByVal e As EventArgs) As Boolean

17 [JScript] protected override function OnBubbleEvent(source : Object, e :

18 EventArgs) : Boolean;

20 *Description*

21 OnCancelCommand

23 [C#] protected virtual void OnCancelCommand(DataListCommandEventArgs e);

24 [C++] protected: virtual void OnCancelCommand(DataListCommandEventArgs\*

25 e);

[VB] Overridable Protected Sub OnCancelCommand(ByVal e As  
DataListCommandEventArgs)

[JScript] protected function OnCancelCommand(e :  
DataListCommandEventArgs);

#### *Description*

Raises the **System.Web.UI.WebControls.DataList.CancelCommand** event. This allows you to provide a custom handler for the event.

Use the

**System.Web.UI.WebControls.DataList.OnCancelCommand(System.Web.UI.  
WebControls.DataListCommandEventArgs)** method to provide a custom  
handler for the **System.Web.UI.WebControls.DataList.CancelCommand** event.  
A **System.Web.UI.WebControls.DataListCommandEventArgs** that contains  
event data.

#### **OnDeleteCommand**

[C#] protected virtual void OnDeleteCommand(DataListCommandEventArgs e);

[C++] protected: virtual void OnDeleteCommand(DataListCommandEventArgs\*  
e);

[VB] Overridable Protected Sub OnDeleteCommand(ByVal e As  
DataListCommandEventArgs)

[JScript] protected function OnDeleteCommand(e :  
DataListCommandEventArgs);

#### *Description*



1       Raises the **System.Web.UI.WebControls.DataList.DeleteCommand**  
2 event. This allows you to provide a custom handler for the event.

3       Use the  
4 **System.Web.UI.WebControls.DataList.OnDeleteCommand(System.Web.UI.**  
5 **WebControls.DataListCommandEventArgs)** method to provide a custom  
6 handler for the **System.Web.UI.WebControls.DataList.DeleteCommand** event.  
7 A **System.Web.UI.WebControls.DataListCommandEventArgs** that contains  
8 event data.

#### 9       OnEditCommand

10  
11 [C#] protected virtual void OnEditCommand(DataListCommandEventArgs e);  
12 [C++] protected: virtual void OnEditCommand(DataListCommandEventArgs\* e);  
13 [VB] Overridable Protected Sub OnEditCommand(ByVal e As  
14 DataListCommandEventArgs)  
15 [JScript] protected function OnEditCommand(e : DataListCommandEventArgs);  
16

#### 17       Description

18       Raises the **System.Web.UI.WebControls.DataList.EditCommand** event.  
19 This allows you to provide a custom handler for the event.

20       Use the  
21 **System.Web.UI.WebControls.DataList.OnEditCommand(System.Web.UI.We**  
22 **bControls.DataListCommandEventArgs)** method to provide a custom handler  
23 for the **System.Web.UI.WebControls.DataList.EditCommand** event. A  
24 **System.Web.UI.WebControls.DataListCommandEventArgs** that contains  
25 event data.

## OnItemCommand

[C#] protected virtual void OnItemCommand(DataListCommandEventArgs e);  
[C++] protected: virtual void OnItemCommand(DataListCommandEventArgs\* e);  
[VB] Overridable Protected Sub OnItemCommand(ByVal e As  
DataListCommandEventArgs)  
[JScript] protected function OnItemCommand(e : DataListCommandEventArgs);

### *Description*

Raises the **System.Web.UI.WebControls.DataList.ItemCommand** event.  
This allows you to provide a custom handler for the event.

Use the  
**System.Web.UI.WebControls.DataList.OnItemCommand(System.Web.UI.WebControls.DataListCommandEventArgs)** method to provide a custom handler for the **System.Web.UI.WebControls.DataList.ItemCommand** event. A **System.Web.UI.WebControls.DataListCommandEventArgs** that contains event data.

## OnItemCreated

[C#] protected virtual void OnItemCreated(DataListItemEventArgs e);  
[C++] protected: virtual void OnItemCreated(DataListItemEventArgs\* e);  
[VB] Overridable Protected Sub OnItemCreated(ByVal e As  
DataListItemEventArgs)  
[JScript] protected function OnItemCreated(e : DataListItemEventArgs);

## *Description*

Raises the **System.Web.UI.WebControls.DataList.ItemCreated** event.

This allows you to provide a custom handler for the event.

Use the

**System.Web.UI.WebControls.DataList.OnItemCreated(System.Web.UI.WebControls.DataListItemEventArgs)** method to provide a custom handler for the **System.Web.UI.WebControls.DataList.ItemCreated** event. A **System.Web.UI.WebControls.DataListItemEventArgs** that contains event data.

## **OnItemDataBound**

[C#] protected virtual void OnItemDataBound(DataListItemEventArgs e);

[C++] protected: virtual void OnItemDataBound(DataListItemEventArgs\* e);

[VB] Overridable Protected Sub OnItemDataBound(ByVal e As DataListItemEventArgs)

[JScript] protected function OnItemDataBound(e : DataListItemEventArgs);

## *Description*

Raises the **System.Web.UI.WebControls.DataList.ItemDataBound** event. This allows you to provide a custom handler for the event.

Use the

**System.Web.UI.WebControls.DataList.OnItemDataBound(System.Web.UI.WebControls.DataListItemEventArgs)** method to provide a custom handler for the **System.Web.UI.WebControls.DataList.ItemDataBound** event. A **System.Web.UI.WebControls.DataListItemEventArgs** that contains event data.

## OnUpdateCommand

```
[C#] protected virtual void OnUpdateCommand(DataListCommandEventArgs e);  
[C++] protected: virtual void OnUpdateCommand(DataListCommandEventArgs*  
e);  
[VB] Overridable Protected Sub OnUpdateCommand(ByVal e As  
DataListCommandEventArgs)  
[JScript] protected function OnUpdateCommand(e :  
DataListCommandEventArgs);
```

### *Description*

Raises the **System.Web.UI.WebControls.DataList.UpdateCommand** event. This allows you to provide a custom handler for the event.

Use the **System.Web.UI.WebControls.DataList.OnUpdateCommand(System.Web.UI.WebControls.DataListCommandEventArgs)** method to provide a custom handler for the **System.Web.UI.WebControls.DataList.UpdateCommand** event. A **System.Web.UI.WebControls.DataListItemEventArgs** that contains event data.

### *PrepareControlHierarchy*

```
[C#] protected override void PrepareControlHierarchy();  
[C++] protected: void PrepareControlHierarchy();  
[VB] Overrides Protected Sub PrepareControlHierarchy()  
[JScript] protected override function PrepareControlHierarchy();
```

1  
2 *Description*

3       RenderContents

4  
5 [C#] protected override void RenderContents(HtmlTextWriter writer);  
6 [C++] protected: void RenderContents(HtmlTextWriter\* writer);  
7 [VB] Overrides Protected Sub RenderContents(ByVal writer As HtmlTextWriter)  
8 [JScript] protected override function RenderContents(writer : HtmlTextWriter);  
9

10 *Description*

11       SaveViewState

12  
13 [C#] protected override object SaveViewState();  
14 [C++] protected: Object\* SaveViewState();  
15 [VB] Overrides Protected Function SaveViewState() As Object  
16 [JScript] protected override function SaveViewState() : Object;  
17

18 *Description*

19       IRepeatInfoUser.GetItemStyle

20  
21 [C#] Style IRepeatInfoUser.GetItemStyle(ListItemType itemType, int  
22 repeatIndex);  
23 [C++] Style\* IRepeatInfoUser::GetItemStyle(ListItemType itemType, int  
24 repeatIndex);  
25 [VB] Function GetItemStyle(ByVal itemType As ListItemType, ByVal

```

1 repeatIndex As Integer) As Style Implements IRepeatInfoUser.GetItemStyle
2 [JScript] function IRepeatInfoUser.GetItemStyle(itemType : ListItemType,
3 repeatIndex : int) : Style;
4         IRepeatInfoUser.RenderItem
5
6 [C#] void IRepeatInfoUser.RenderItem(ListItemType itemType, int repeatIndex,
7 RepeatInfo repeatInfo, HtmlTextWriter writer);
8 [C++] void IRepeatInfoUser::RenderItem(ListItemType itemType, int
9 repeatIndex, RepeatInfo* repeatInfo, HtmlTextWriter* writer);
10 [VB] Sub RenderItem(ByVal itemType As ListItemType, ByVal repeatIndex As
11 Integer, ByVal repeatInfo As RepeatInfo, ByVal writer As HtmlTextWriter)
12 Implements IRepeatInfoUser.RenderItem
13 [JScript] function IRepeatInfoUser.RenderItem(itemType : ListItemType,
14 repeatIndex : int, repeatInfo : RepeatInfo, writer : HtmlTextWriter);
15         ViewState
16
17 [C#] protected override void ViewState();
18 [C++] protected: void ViewState();
19 [VB] Overrides Protected Sub ViewState()
20 [JScript] protected override function ViewState();
21
22 Description

```

Marks the starting point to begin tracking and saving changes to the control  
 as part of the control viewstate.

DataListCommandEventArgs class (System.Web.UI.WebControls)

## TrackViewState

### *Description*

Provides data for the **System.Web.UI.WebControls.DataList.CancelCommand** , **System.Web.UI.WebControls.DataList.DeleteCommand** , **System.Web.UI.WebControls.DataList.EditCommand** , **System.Web.UI.WebControls.DataList.ItemCommand** , and **System.Web.UI.WebControls.DataList.UpdateCommand** events of the **System.Web.UI.WebControls.DataList** control. This class cannot be inherited.

The **System.Web.UI.WebControls.DataList.CancelCommand** event is raised when the **Cancel** button for an item in the **System.Web.UI.WebControls.DataList** control is clicked.

**DataListCommandEventArgs**

### *Example Syntax:*

**TrackViewState**

```
[C#] public DataListCommandEventArgs(DataListItem item, object  
commandSource, CommandEventArgs originalArgs);
```

```
[C++] public: DataListCommandEventArgs(DataListItem* item, Object*  
commandSource, CommandEventArgs* originalArgs);
```

```
[VB] Public Sub New(ByVal item As DataListItem, ByVal commandSource As  
Object, ByVal originalArgs As CommandEventArgs)
```

```
[JScript] public function DataListCommandEventArgs(item : DataListItem,
```

1 commandSource : Object, originalArgs : CommandEventArgs);

2  
3 *Description*

4       Initializes a new instance of the  
5 **System.Web.UI.WebControls.DataListCommandEventArgs** class.

6       Use this constructor to create and initialize a new instance of the  
7 **System.Web.UI.WebControls.DataListCommandEventArgs** class. The  
8 selected item from the **System.Web.UI.WebControls.DataList**. The source of  
9 the command. A **System.Web.UI.WebControls.CommandEventArgs** that  
10 contains the original event data.

11       CommandArgument

12       CommandName

13       CommandSource

14       TrackViewState

15  
16  
17 *Description*

18       Gets the source of the command.

19       Use the  
20 **System.Web.UI.WebControls.DataListCommandEventArgs.CommandSource**  
21 e property to determine the command source that raised the event. This property is  
22 commonly used to determine which command raises the event. You can then take  
23 appropriate action, based on the command.

24       Item

25       TrackViewState



```

1
2 [C#] public DataListItem Item {get;}
3 [C++] public: __property DataListItem* get_Item();
4 [VB] Public ReadOnly Property Item As DataListItem
5 [JScript] public function get Item() : DataListItem;
6

```

### *Description*

Gets the item containing the command source in the **System.Web.UI.WebControls.DataList** control.

Use the **System.Web.UI.WebControls.DataListCommandEventArgs.Item** property to access information about the selected item in the **System.Web.UI.WebControls.DataList** control.

**DataListCommandEventHandler** delegate (System.Web.UI.WebControls)  
**ToString**

### *Description*

Represents the method that will handle the **System.Web.UI.WebControls.DataList.CancelCommand** , **System.Web.UI.WebControls.DataList.DeleteCommand** , **System.Web.UI.WebControls.DataList.EditCommand** , **System.Web.UI.WebControls.DataList.ItemCommand** , and **System.Web.UI.WebControls.DataList.UpdateCommand** events of a **System.Web.UI.WebControls.DataList** control. The source of the event. A

1 **System.Web.UI.WebControls.DataListCommandEventArgs** that contains the  
2 event data.

3 The **System.Web.UI.WebControls.DataList.CancelCommand** event is  
4 raised when the **Cancel** button for an item in the  
5 **System.Web.UI.WebControls.DataList** control is clicked.

6 **DataListItem** class (System.Web.UI.WebControls)

7 **ToString**

8  
9  
10 *Description*

11 Represents an item in the **System.Web.UI.WebControls.DataList** control.

12 A **System.Web.UI.WebControls.DataListItem** object represents an item  
13 in the **System.Web.UI.WebControls.DataList** control, such as the heading  
14 section, the footer section, or a data item.

15 **DataListItem**

16 *Example Syntax:*

17 **ToString**

18  
19 [C#] public **DataListItem**(int itemIndex, ListItemType itemType);

20 [C++] public: **DataListItem**(int itemIndex, ListItemType itemType);

21 [VB] Public Sub New(ByVal itemIndex As Integer, ByVal itemType As  
22 ListItemType)

23 [JScript] public function **DataListItem**(itemIndex : int, itemType : ListItemType);

24  
25 *Description*

1        Initializes a new instance of the  
 2        **System.Web.UI.WebControls.DataListItem** class.  
 3        Use this constructor to create and initialize a new instance of the  
 4        **System.Web.UI.WebControls.DataListItem** class. The index of the item in the  
 5        **System.Web.UI.WebControls.DataList** control from the  
 6        **System.Web.UI.WebControls.DataList.Items** collection. One of the  
 7        **System.Web.UI.WebControls.ListItemType** values.

8        AccessKey  
 9        Attributes  
 10       BackColor  
 11       BorderColor  
 12       BorderStyle  
 13       BorderWidth  
 14       ChildControlsCreated  
 15       ClientID  
 16       Context  
 17       Controls  
 18       ControlStyle  
 19       ControlStyleCreated  
 20       CssClass  
 21       DataItem  
 22       ToString

25       *Description*

Gets or sets a data item associated with the  
**System.Web.UI.WebControls.DataListItem** object in the  
**System.Web.UI.WebControls.DataList** control.

Use the **System.Web.UI.WebControls.DataListItem.DataItem** property  
to specify or determine the properties of a data item associated with the  
**System.Web.UI.WebControls.DataListItem** object in the  
**System.Web.UI.WebControls.DataList** control.

Enabled

EnableViewState

Events

Font

ForeColor

HasChildViewState

Height

ID

IsTrackingViewState

ItemIndex

ToString

### *Description*

Gets the index of the **System.Web.UI.WebControls.DataListItem** object  
from from the **System.Web.UI.WebControls.DataList.Items** collection of the  
control.

1        Use the **System.Web.UI.WebControls.DataListItem.ItemIndex** property  
2 to determine the index number of the  
3 **System.Web.UI.WebControls.DataListItem** object from the  
4 **System.Web.UI.WebControls.DataList.Items** collection.

5        ItemType

6        ToString

7  
8 [C#] public virtual ListItemType ItemType {get;}

9 [C++] public: \_\_property virtual ListItemType get\_ItemType();

10 [VB] Overridable Public ReadOnly Property ItemType As ListItemType

11 [JScript] public function get ItemType() : ListItemType;

12  
13 *Description*

14        Gets the type of the item represented by the  
15 **System.Web.UI.WebControls.DataListItem** object in the  
16 **System.Web.UI.WebControls.DataList** control.

17        Use the **System.Web.UI.WebControls.DataListItem.ItemType** property  
18 to determine the type of the item represented by the  
19 **System.Web.UI.WebControls.DataGridItem** object. The following table lists  
20 the various item types.

21        NamingContainer

22        Page

23        Parent

24        Site

25        Style

1 TabIndex  
 2 TagKey  
 3 TagName  
 4 TemplateSourceDirectory  
 5 ToolTip  
 6 UniqueID  
 7 ViewState  
 8 ViewStateIgnoresCase  
 9 Visible  
 10 Width  
 11 CreateControlStyle

12  
 13 [C#] protected override Style CreateControlStyle();  
 14 [C++] protected: Style\* CreateControlStyle();  
 15 [VB] Overrides Protected Function CreateControlStyle() As Style  
 16 [JScript] protected override function CreateControlStyle() : Style;

17  
 18 *Description*

19 OnBubbleEvent

20  
 21 [C#] protected override bool OnBubbleEvent(object source, EventArgs e);  
 22 [C++] protected: bool OnBubbleEvent(Object\* source, EventArgs\* e);  
 23 [VB] Overrides Protected Function OnBubbleEvent(ByVal source As Object,  
 24 ByVal e As EventArgs) As Boolean  
 25 [JScript] protected override function OnBubbleEvent(source : Object, e :

EventArgs) : Boolean;

### Description

#### RenderItem

[C#] public virtual void RenderItem(HtmlTextWriter writer, bool extractRows, bool tableLayout);

[C++] public: virtual void RenderItem(HtmlTextWriter\* writer, bool extractRows, bool tableLayout);

[VB] Overridable Public Sub RenderItem(ByVal writer As HtmlTextWriter, ByVal extractRows As Boolean, ByVal tableLayout As Boolean)

[JScript] public function RenderItem(writer : HtmlTextWriter, extractRows : Boolean, tableLayout : Boolean);

### Description

Displays the **System.Web.UI.WebControls.DataListItem** object on the client.

Use the

**System.Web.UI.WebControls.DataListItem.RenderItem(System.Web.UI.HtmlTextWriter, System.Boolean, System.Boolean)** method to display the **System.Web.UI.WebControls.DataListItem** object on the client. A **System.Web.UI.HtmlTextWriter** object that contains the output stream for rendering on the client. **true** to extract rows; otherwise **false**. **true** to display as a table; otherwise **false**.

#### SetItemType

1  
2 [C#] protected internal virtual void SetItemType(ListItemType itemType);  
3 [C++] protected public: virtual void SetItemType(ListItemType itemType);  
4 [VB] Overridable Protected Friend Dim Sub SetItemType(ByVal itemType As  
5 ListItemType)  
6 [JScript] package function SetItemType(itemType : ListItemType);  
7

8 *Description*

9 DataListItemCollection class (System.Web.UI.WebControls)

10 TrackViewState  
11

12  
13 *Description*

14 Represents the collection of **System.Web.UI.WebControls.DataListItem**  
15 objects in the **System.Web.UI.WebControls.DataList** control. This class cannot  
16 be inherited.

17 The **System.Web.UI.WebControls.DataListItemCollection** class  
18 represents a collection of **System.Web.UI.WebControls.DataListItem** objects,  
19 which in turn represent the data items in a  
20 **System.Web.UI.WebControls.DataList** control. To programmatically retrieve  
21 **System.Web.UI.WebControls.DataListItem** objects from a  
22 **System.Web.UI.WebControls.DataList** control, use one of the following  
23 methods: Use the indexer to get a single  
24 **System.Web.UI.WebControls.DataListItem** object from the collection, using  
25 array notation.



DataListItemCollection

*Example Syntax:*

TrackViewState

[C#] public DataListItemCollection(ArrayList items);

[C++] public: DataListItemCollection(ArrayList\* items);

[VB] Public Sub New(ByVal items As ArrayList)

[JScript] public function DataListItemCollection(items : ArrayList);

### *Description*

Initializes a new instance of the

**System.Web.UI.WebControls.DataListItemCollection** class.

Use this constructor to create and initialize a new instance of the

**System.Web.UI.WebControls.DataListItemCollection** class. A

**System.Collections.ArrayList** object that contains the items with which to initialize the collection.

Count

TrackViewState

[C#] public int Count {get;}

[C++] public: \_\_property int get\_Count();

[VB] Public ReadOnly Property Count As Integer

[JScript] public function get Count() : int;

### *Description*

Gets the number of **System.Web.UI.WebControls.DataListItem** objects in the collection.

Use the **System.Web.UI.WebControls.DataListItemCollection.Count** property to determine the number of **System.Web.UI.WebControls.DataListItem** objects in the **System.Web.UI.WebControls.DataListItemCollection** collection. The **System.Web.UI.WebControls.DataListItemCollection.Count** property is commonly used when iterating through the collection to determine the upper bound of the collection.

**IsReadOnly**

**TrackViewState**

[C#] public bool IsReadOnly {get;}

[C++] public: \_\_property bool get\_IsReadOnly();

[VB] Public ReadOnly Property IsReadOnly As Boolean

[JScript] public function get IsReadOnly() : Boolean;

### *Description*

Gets a value that indicates whether the **System.Web.UI.WebControls.DataListItem** objects in the **System.Web.UI.WebControls.DataListItemCollection** can be modified.

This property always returns **false** to indicate that the **System.Web.UI.WebControls.DataListItemCollection** can be written to in all cases.

**IsSynchronized**

## TrackViewState

[C#] public bool IsSynchronized {get;}

[C++] public: \_\_property bool get\_IsSynchronized();

[VB] Public ReadOnly Property IsSynchronized As Boolean

[JScript] public function get IsSynchronized() : Boolean;

### *Description*

Gets a value indicating whether access to the **System.Web.UI.WebControls.DataListItemCollection** is synchronized (thread-safe).

This property is derived from the **System.Collections.ICollection** class and is overridden to always return **false**.

### *Item*

## TrackViewState

[C#] public DataListItem this[int index] {get;}

[C++] public: \_\_property DataListItem\* get\_Item(int index);

[VB] Public Default ReadOnly Property Item(ByVal index As Integer) As

DataListItem

[JScript] returnValue = DataListItemCollectionObject.Item(index);

### *Description*

Gets a **System.Web.UI.WebControls.DataListItem** object at the specified index in the collection.

1        Use this indexer to get a **System.Web.UI.WebControls.DataListItem**  
2 object from the **System.Web.UI.WebControls.DataListItemCollection** at the  
3 specified index using array notation. The index of the  
4 **System.Web.UI.WebControls.DataListItem** in the collection to retrieve.

5        SyncRoot

6        TrackViewState

7  
8 [C#] public object SyncRoot {get;}

9 [C++] public: \_\_property Object\* get\_SyncRoot();

10 [VB] Public ReadOnly Property SyncRoot As Object

11 [JScript] public function get SyncRoot() : Object;

12  
13 *Description*

14        Gets the object that can be used to synchronize access to the  
15 **System.Web.UI.WebControls.DataListItemCollection** collection.

16        The object returned in this implementation is the  
17 **System.Web.UI.WebControls.DataListItemCollection** object itself.

18        CopyTo

19  
20 [C#] public void CopyTo(Array array, int index);

21 [C++] public: \_\_sealed void CopyTo(Array\* array, int index);

22 [VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As  
23 Integer)

24 [JScript] public function CopyTo(array : Array, index : int);

1  
2 *Description*

3       Copies all the items from this  
4 **System.Web.UI.WebControls.DataListItemCollection** collection to the  
5 specified **System.Array** object, starting at the specified index in the  
6 **System.Array** object.

7       Use this method to copy the contents of the  
8 **System.Web.UI.WebControls.DataListItemCollection** collection into the  
9 specified **System.Array** object, starting at the specified index. A zero-based  
10 **System.Array** object that receives the copied items from the  
11 **System.Web.UI.WebControls.DataListItemCollection** collection. The first  
12 position in the specified **System.Array** object to receive the copied contents.

13       GetEnumerator

14  
15 [C#] public IEnumerator GetEnumerator();  
16 [C++] public: \_\_sealed IEnumerator\* GetEnumerator();  
17 [VB] NotOverridable Public Function GetEnumerator() As IEnumerator  
18 [JScript] public function GetEnumerator() : IEnumerator;  
19

20 *Description*

21       Returns a **System.Collections.IEnumerator** interface that contains all  
22 **System.Web.UI.WebControls.DataListItem** objects in the  
23 **System.Web.UI.WebControls.DataListItemCollection** .

24 *Return Value:* A **System.Collections.IEnumerator** interface that contains all  
25

1 **System.Web.UI.WebControls.DataListItem** objects in the  
2 **System.Web.UI.WebControls.DataListItemCollection** .

3       Use this method to create a **System.Collections.IEnumerator** that can be  
4 easily iterated through to get each item in the  
5 **System.Web.UI.WebControls.DataListItemCollection** collection.

6       DataListItemEventArgs class (System.Web.UI.WebControls)

7       ToString

8  
9  
10 *Description*

11       Provides data for the  
12 **System.Web.UI.WebControls.DataList.ItemCreated** and  
13 **System.Web.UI.WebControls.DataList.ItemDataBound** events of a  
14 **System.Web.UI.WebControls.DataList** control. This class cannot be inherited.

15       The **System.Web.UI.WebControls.DataList.ItemCreated** event is raised  
16 when an item in the **System.Web.UI.WebControls.DataList** control is created.

17       DataListItemEventArgs

18 *Example Syntax:*

19       ToString

20  
21 [C#] public DataListItemEventArgs(DataListItem item);

22 [C++] public: DataListItemEventArgs(DataListItem\* item);

23 [VB] Public Sub New(ByVal item As DataListItem)

24 [JScript] public function DataListItemEventArgs(item : DataListItem);

1  
2 *Description*

3       Initializes a new instance of the  
4 **System.Web.UI.WebControls.DataListItemEventArgs** class.

5       Use this constructor to create and initialize a new instance of the  
6 **System.Web.UI.WebControls.DataListItemEventArgs** class. A  
7 **System.Web.UI.WebControls.DataListItem** object that represents an item in the  
8 **System.Web.UI.WebControls.DataList** control.

9       Item

10       ToString

11  
12 [C#] public DataListItem Item {get;}

13 [C++] public: \_\_property DataListItem\* get\_Item();

14 [VB] Public ReadOnly Property Item As DataListItem

15 [JScript] public function get Item() : DataListItem;

16  
17 *Description*

18       Gets the referenced item in the **System.Web.UI.WebControls.DataList**  
19 control when the event is raised.

20       The items in the **System.Web.UI.WebControls.DataList** control are  
21 referenced as each item is created or bound to the control. Use this property to  
22 programmatically access the referenced item in the  
23 **System.Web.UI.WebControls.DataList** control when the  
24 **System.Web.UI.WebControls.DataList.ItemCreated** or  
25 **System.Web.UI.WebControls.DataList.ItemDataBound** event is raised.

1 DataListItemEventHandler delegate (System.Web.UI.WebControls)

2 ToString

3  
4  
5 *Description*

6 Represents the method that will handle the

7 **System.Web.UI.WebControls.DataList.ItemCreated** and

8 **System.Web.UI.WebControls.DataList.ItemDataBound** events of the

9 **System.Web.UI.WebControls.DataList** control. The source of the event. A

10 **System.Web.UI.WebControls.DataListItemEventArgs** that contains the event  
11 data.

12 The **System.Web.UI.WebControls.DataList.ItemCreated** event is raised  
13 when an item in the **System.Web.UI.WebControls.DataList** control is created.

14 DayNameFormat enumeration (System.Web.UI.WebControls)

15 ToString

16  
17  
18 *Description*

19 Specifies the display format for the days of the week on a

20 **System.Web.UI.WebControls.Calendar** control.

21 The **System.Web.UI.WebControls.DayNameFormat** enumeration  
22 represents the display formats for the days of the week on a

23 **System.Web.UI.WebControls.Calendar** control.

24 ToString



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#] public const DayNameFormat FirstLetter;  
[C++] public: const DayNameFormat FirstLetter;  
[VB] Public Const FirstLetter As DayNameFormat  
[JScript] public var FirstLetter : DayNameFormat;
```

*Description*

The days of the week displayed with just the first letter. For example, T.  
ToString

```
[C#] public const DayNameFormat FirstTwoLetters;  
[C++] public: const DayNameFormat FirstTwoLetters;  
[VB] Public Const FirstTwoLetters As DayNameFormat  
[JScript] public var FirstTwoLetters : DayNameFormat;
```

*Description*

The days of the week displayed with just the first two letters. For example,  
Tu.  
ToString

```
[C#] public const DayNameFormat Full;  
[C++] public: const DayNameFormat Full;  
[VB] Public Const Full As DayNameFormat  
[JScript] public var Full : DayNameFormat;
```

TOP-00000000

1  
2 *Description*

3 The days of the week displayed in full format. For example, Tuesday.

4 ToString

5  
6 [C#] public const DayNameFormat Short;

7 [C++] public: const DayNameFormat Short;

8 [VB] Public Const Short As DayNameFormat

9 [JScript] public var Short : DayNameFormat;

10  
11 *Description*

12 The days of the week displayed in abbreviated format. For example, Tues.

13 DayRenderEventArgs class (System.Web.UI.WebControls)

14 ToString

15  
16  
17 *Description*

18 Provides data for the

19 **System.Web.UI.WebControls.Calendar.DayRender** event of the

20 **System.Web.UI.WebControls.Calendar** control. This class cannot be inherited.

21 Although data binding is not supported for the

22 **System.Web.UI.WebControls.Calendar** control, it is possible to modify the

23 content and formatting of the individual date cells. Before the

24 **System.Web.UI.WebControls.Calendar** control is displayed on the Web page, it

25 creates and assembles the components that make up the control. The

**System.Web.UI.WebControls.Calendar.DayRender** event is raised when each date cell in **System.Web.UI.WebControls.Calendar** control is created. You can control the contents and formatting of a date cell when it is created by providing code in the event handler for the

**System.Web.UI.WebControls.Calendar.DayRender** event.

**DayRenderEventArgs**

*Example Syntax:*

**ToString**

[C#] public DayRenderEventArgs(TableCell cell, CalendarDay day);

[C++] public: DayRenderEventArgs(TableCell\* cell, CalendarDay\* day);

[VB] Public Sub New(ByVal cell As TableCell, ByVal day As CalendarDay)

[JScript] public function DayRenderEventArgs(cell : TableCell, day :

CalendarDay);

### *Description*

Initializes a new instance of the **System.Web.UI.WebControls.DayRenderEventArgs** class.

Use this constructor to create and initialize a new instance of the **System.Web.UI.WebControls.DayRenderEventArgs** class. A **System.Web.UI.WebControls.TableCell** object that represents a cell in the **System.Web.UI.WebControls.Calendar** control. A **System.Web.UI.WebControls.CalendarDay** object that represents the day to render in the **System.Web.UI.WebControls.Calendar** control.

**Cell**

ToString

```
[C#] public TableCell Cell {get;}
[C++] public: __property TableCell* get_Cell();
[VB] Public ReadOnly Property Cell As TableCell
[JScript] public function get Cell() : TableCell;
```

*Description*

Gets the **System.Web.UI.WebControls.TableCell** object that represents the cell being rendered in the **System.Web.UI.WebControls.Calendar** control.

Use the **System.Web.UI.WebControls.DayRenderEventArgs.Cell** property to programmatically control the cell being rendered in the **System.Web.UI.WebControls.Calendar** control.

Day

ToString

```
[C#] public CalendarDay Day {get;}
[C++] public: __property CalendarDay* get_Day();
[VB] Public ReadOnly Property Day As CalendarDay
[JScript] public function get Day() : CalendarDay;
```

*Description*

Gets the **System.Web.UI.WebControls.CalendarDay** that represents the day being rendered in the **System.Web.UI.WebControls.Calendar** control.

1        Use the **System.Web.UI.WebControls.DayRenderEventArgs.Day**  
2        property to get information about the day being rendered in the  
3        **System.Web.UI.WebControls.Calendar** control.

4        DayRenderEventHandler delegate (System.Web.UI.WebControls)  
5        ToString

6  
7  
8        *Description*

9        Represents the method that will handle the  
10       **System.Web.UI.WebControls.Calendar.DayRender** event of the  
11       **System.Web.UI.WebControls.Calendar** control. The source of the event. A  
12       **System.Web.UI.WebControls.DayRenderEventArgs** that contains the event  
13       data.

14       Although data binding is not supported for the  
15       **System.Web.UI.WebControls.Calendar** control, it is possible to modify the  
16       content and formatting of the individual date cells. Before the  
17       **System.Web.UI.WebControls.Calendar** control is displayed on the Web page, it  
18       creates and assembles the components that make up the control. The  
19       **System.Web.UI.WebControls.Calendar.DayRender** event is raised when each  
20       date cell in **System.Web.UI.WebControls.Calendar** control is created. You can  
21       control the contents and formatting of a date cell when it is created by providing  
22       code in the event handler for the  
23       **System.Web.UI.WebControls.Calendar.DayRender** event. For additional  
24       information on customizing the contents of a date cell, see

1 **System.Web.UI.WebControls.Calendar.OnDayRender(System.Web.UI.Web**  
2 **Controls.TableCell,System.Web.UI.WebControls.CalendarDay) .**

3       DropDownList class (System.Web.UI.WebControls)

4       ToString

5  
6  
7 *Description*

8       Represents a control that allows the user to select a single item from a drop-  
9 down list.

10       Use the **System.Web.UI.WebControls.DropDownList** control to create a  
11 single selection drop-down list control. You can control the appearance of the  
12 **System.Web.UI.WebControls.DropDownList** control by setting the  
13 **System.Web.UI.WebControls.DropDownList.BorderColor** ,  
14 **System.Web.UI.WebControls.DropDownList.BorderStyle** , and  
15 **System.Web.UI.WebControls.DropDownList.BorderWidth** properties.

16       DropDownList

17       *Example Syntax:*

18       ToString

19  
20 [C#] public DropDownList();

21 [C++] public: DropDownList();

22 [VB] Public Sub New()

23 [JScript] public function DropDownList();

24  
25 *Description*

1        Initializes a new instance of the  
2        **System.Web.UI.WebControls.DropDownList** class.

3        Use this constructor to create and initialize a new instance of the  
4        **System.Web.UI.WebControls.DropDownList** class.

5        AccessKey

6        Attributes

7        AutoPostBack

8        BackColor

9        BorderColor

10       ToString

11  
12  
13        *Description*

14        Gets or sets the border color of the control.

15        The **System.Web.UI.WebControls.DropDownList.BorderColor**  
16        property is inherited from the **System.Web.UI.WebControls.WebControl** class  
17        and is not applicable to the **System.Web.UI.WebControls.DropDownList**  
18        control.

19        BorderStyle

20        ToString

21  
22        [C#] public override BorderStyle BorderStyle {get; set;}

23        [C++] public: \_\_property virtual BorderStyle get\_BorderStyle();public:

24        \_\_property virtual void set\_BorderStyle(BorderStyle);

25        [VB] Overrides Public Property BorderStyle As BorderStyle

[JScript] public function get BorderStyle() : BorderStyle;public function set  
BorderStyle(BorderStyle);

*Description*

Gets or sets the border style of the control.

The **System.Web.UI.WebControls.DropDownList.BorderStyle** property is inherited from the **System.Web.UI.WebControls.WebControl** class and is not applicable to the **System.Web.UI.WebControls.DropDownList** control.

BorderWidth

ToString

[C#] public override Unit BorderWidth {get; set;}

[C++] public: \_\_property virtual Unit get\_ BorderWidth();public: \_\_property  
virtual void set\_ BorderWidth(Unit);

[VB] Overrides Public Property BorderWidth As Unit

[JScript] public function get BorderWidth() : Unit;public function set  
BorderWidth(Unit);

*Description*

Gets or sets the border width for the control.

The **System.Web.UI.WebControls.DropDownList.BorderWidth** property is inherited from the **System.Web.UI.WebControls.WebControl** class and is not applicable to the **System.Web.UI.WebControls.DropDownList** control.

ChildControlsCreated



700740-00000000

1	ClientID
2	Context
3	Controls
4	ControlStyle
5	ControlStyleCreated
6	CssClass
7	DataMember
8	DataSource
9	DataTextField
10	DataTextFormatString
11	DataValueField
12	Enabled
13	EnableViewState
14	Events
15	Font
16	ForeColor
17	HasChildViewState
18	Height
19	ID
20	IsTrackingViewState
21	Items
22	NamingContainer
23	Page
24	Parent
25	SelectedIndex

ToString

*Description*

Gets or sets the index of the selected item in the **System.Web.UI.WebControls.DropDownList** control.

Use the **System.Web.UI.WebControls.DropDownList.SelectedIndex** property to programmatically specify or determine the index of the selected item from the **System.Web.UI.WebControls.DropDownList** control. An item is always selected in the **System.Web.UI.WebControls.DropDownList** control. You cannot deselect every item in the list at the same time.

SelectedItem

Site

Style

TabIndex

TagKey

TagName

TemplateSourceDirectory

ToolTip

ToString

*Description*

Gets or sets the ToolTip text displayed when the mouse pointer rests over the control.

1 The **System.Web.UI.WebControls.DropDownList.ToolTip** property is  
2 inherited from the **System.Web.UI.WebControls.WebControl** class and is not  
3 applicable to the **System.Web.UI.WebControls.DropDownList** control.

4 UniqueID

5 ViewState

6 ViewStateIgnoresCase

7 Visible

8 Width

9 AddAttributesToRender

10  
11 [C#] protected override void AddAttributesToRender(HtmlTextWriter writer);

12 [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);

13 [VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As  
14 HtmlTextWriter)

15 [JScript] protected override function AddAttributesToRender(writer :  
16 HtmlTextWriter);

17  
18 *Description*

19 Adds the properties of the **System.Web.UI.WebControls.DropDownList**  
20 control to the output stream for rendering on the client. The output stream for  
21 rendering on the client.

22 CreateControlCollection

23  
24 [C#] protected override ControlCollection CreateControlCollection();

25 [C++] protected: ControlCollection\* CreateControlCollection();

1 [VB] Overrides Protected Function CreateControlCollection() As  
2 ControlCollection

3 [JScript] protected override function CreateControlCollection() :  
4 ControlCollection;

5  
6 *Description*

7  
8 RenderContents

9  
10 [C#] protected override void RenderContents(HtmlTextWriter writer);  
11 [C++] protected: void RenderContents(HtmlTextWriter\* writer);  
12 [VB] Overrides Protected Sub RenderContents(ByVal writer As HtmlTextWriter)  
13 [JScript] protected override function RenderContents(writer : HtmlTextWriter);  
14

15 *Description*

16 Displays the **System.Web.UI.WebControls.DropDownList** control on the  
17 client. The output stream for rendering on the client.

18 IPostBackDataHandler.LoadPostData

19  
20 [C#] bool IPostBackDataHandler.LoadPostData(string postDataKey,  
21 NameValueCollection postCollection);  
22 [C++] bool IPostBackDataHandler::LoadPostData(String\* postDataKey,  
23 NameValueCollection\* postCollection);  
24 [VB] Function LoadPostData(ByVal postDataKey As String, ByVal  
25 postCollection As NameValueCollection) As Boolean Implements

1 IPostBackDataHandler.LoadPostData

2 [JScript] function IPostBackDataHandler.LoadPostData(postDataKey : String,  
3 postCollection : NameValueCollection) : Boolean;

4 IPostBackDataHandler.RaisePostDataChangedEvent

6 [C#] void IPostBackDataHandler.RaisePostDataChangedEvent();

7 [C++] void IPostBackDataHandler::RaisePostDataChangedEvent();

8 [VB] Sub RaisePostDataChangedEvent() Implements

9 IPostBackDataHandler.RaisePostDataChangedEvent

10 [JScript] function IPostBackDataHandler.RaisePostDataChangedEvent();

11 EditCommandColumn class (System.Web.UI.WebControls)

12 TrackViewState

### 15 *Description*

16 A special column type for the **System.Web.UI.WebControls.DataGrid**  
17 control that contains the **Edit** command buttons for editing data items in each row.

18 Use the **System.Web.UI.WebControls.EditCommandColumn** class to  
19 create a special column for the **System.Web.UI.WebControls.DataGrid** control  
20 that contains the **Edit** , **Update** , and **Cancel** command buttons for each data row  
21 in the grid. These buttons allow you to edit the values of a row in the

22 **System.Web.UI.WebControls.DataGrid** control.

23 EditCommandColumn

24 *Example Syntax:*

25 TrackViewState

1  
2 [C#] public EditCommandColumn();

3 [C++] public: EditCommandColumn();

4 [VB] Public Sub New()

5 [JScript] public function EditCommandColumn();

6  
7 *Description*

8       Initializes a new instance of the  
9 **System.Web.UI.WebControls.EditCommandColumn** class.

10       Use this constructor to create and initialize a new instance of the  
11 **System.Web.UI.WebControls.EditCommandColumn** class.

12       ButtonType

13       TrackViewState

14  
15 [C#] public virtual ButtonColumnType ButtonType {get; set;}

16 [C++] public: \_\_property virtual ButtonColumnType get\_ButtonType();public:  
17 \_\_property virtual void set\_ButtonType(ButtonColumnType);

18 [VB] Overridable Public Property ButtonType As ButtonColumnType

19 [JScript] public function get ButtonType() : ButtonColumnType;public function  
20 set ButtonType(ButtonColumnType);

21  
22 *Description*

23       Gets or sets the button type for the column.

1 Use the

2 **System.Web.UI.WebControls.EditCommandColumn.ButtonType** property to  
3 specify whether the buttons in the column display as push buttons or hyperlinks.

4 CancelText

5 TrackViewState

6  
7 [C#] public virtual string CancelText {get; set;}

8 [C++] public: \_\_property virtual String\* get\_CancelText();public: \_\_property  
9 virtual void set\_CancelText(String\*);

10 [VB] Overridable Public Property CancelText As String

11 [JScript] public function get CancelText() : String;public function set  
12 CancelText(String);

13  
14 *Description*

15 Gets or sets the text to display for the **Cancel** command button in the  
16 **System.Web.UI.WebControls.EditCommandColumn** .

17 Use the

18 **System.Web.UI.WebControls.EditCommandColumn.CancelText** property to  
19 specify the text to display for the **Cancel** command button in the  
20 **System.Web.UI.WebControls.EditCommandColumn** .

21 DesignMode

22 EditText

23 TrackViewState

1  
2  
3 *Description*

4 Gets or sets the text to display for the **Edit** command button in the  
5 **System.Web.UI.WebControls.EditCommandColumn** .

6 Use the **System.Web.UI.WebControls.EditCommandColumn.EditText**  
7 property to specify the text to display for the **Edit** command button in the  
8 **System.Web.UI.WebControls.EditCommandColumn** .

9 FooterStyle

10 FooterText

11 HeaderImageUrl

12 HeaderStyle

13 HeaderText

14 IsTrackingViewState

15 ItemStyle

16 Owner

17 SortExpression

18 UpdateText

19 TrackViewState

20  
21  
22 *Description*

23 Gets or sets the text to display for the **Update** command button in the  
24 **System.Web.UI.WebControls.EditCommandColumn** .



Use the

**System.Web.UI.WebControls.EditCommandColumn.UpdateText** property to specify the text to display for the **Update** command button in the **System.Web.UI.WebControls.EditCommandColumn**.

ViewState

Visible

InitializeCell

[C#] public override void InitializeCell(TableCell cell, int columnIndex, ListItemType itemType);

[C++] public: void InitializeCell(TableCell\* cell, int columnIndex, ListItemType itemType);

[VB] Overrides Public Sub InitializeCell(ByVal cell As TableCell, ByVal columnIndex As Integer, ByVal itemType As ListItemType)

[JScript] public override function InitializeCell(cell : TableCell, columnIndex : int, itemType : ListItemType);

*Description*

Initializes a cell within the column. A **System.Web.UI.WebControls.TableCell** that contains information about the cell to initialize. The column number where the cell is located. One of the **System.Web.UI.WebControls.ListItemType** values.

FirstDayOfWeek enumeration (System.Web.UI.WebControls)

TrackViewState

1  
2  
3 *Description*

4 Specifies the day to display as the first day of the week on the  
5 **System.Web.UI.WebControls.Calendar** control.

6 The **System.Web.UI.WebControls.FirstDayOfWeek** enumeration  
7 represents the values that specify which day to display as the first day of the week  
8 on the **System.Web.UI.WebControls.Calendar** control.

9 TrackViewState

10  
11 [C#] public const FirstDayOfWeek Default;  
12 [C++] public: const FirstDayOfWeek Default;  
13 [VB] Public Const Default As FirstDayOfWeek  
14 [JScript] public var Default : FirstDayOfWeek;  
15

16 *Description*

17 The first day of the week is specified by the system settings.

18 TrackViewState

19  
20 [C#] public const FirstDayOfWeek Friday;  
21 [C++] public: const FirstDayOfWeek Friday;  
22 [VB] Public Const Friday As FirstDayOfWeek  
23 [JScript] public var Friday : FirstDayOfWeek;  
24

25 *Description*

1 The first day of the week is Friday.

2 TrackViewState

3

4 [C#] public const FirstDayOfWeek Monday;

5 [C++] public: const FirstDayOfWeek Monday;

6 [VB] Public Const Monday As FirstDayOfWeek

7 [JScript] public var Monday : FirstDayOfWeek;

8

9 *Description*

10 The first day of the week is Monday.

11 TrackViewState

12

13 [C#] public const FirstDayOfWeek Saturday;

14 [C++] public: const FirstDayOfWeek Saturday;

15 [VB] Public Const Saturday As FirstDayOfWeek

16 [JScript] public var Saturday : FirstDayOfWeek;

17

18 *Description*

19 The first day of the week is Saturday.

20 TrackViewState

21

22 [C#] public const FirstDayOfWeek Sunday;

23 [C++] public: const FirstDayOfWeek Sunday;

24 [VB] Public Const Sunday As FirstDayOfWeek

25 [JScript] public var Sunday : FirstDayOfWeek;

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

*Description*

The first day of the week is Sunday.

TrackViewState

```
[C#] public const FirstDayOfWeek Thursday;  
[C++] public: const FirstDayOfWeek Thursday;  
[VB] Public Const Thursday As FirstDayOfWeek  
[JScript] public var Thursday : FirstDayOfWeek;
```

*Description*

The first day of the week is Thursday.

TrackViewState

```
[C#] public const FirstDayOfWeek Tuesday;  
[C++] public: const FirstDayOfWeek Tuesday;  
[VB] Public Const Tuesday As FirstDayOfWeek  
[JScript] public var Tuesday : FirstDayOfWeek;
```

*Description*

The first day of the week is Tuesday.

TrackViewState

```
[C#] public const FirstDayOfWeek Wednesday;  
[C++] public: const FirstDayOfWeek Wednesday;
```

1 [VB] Public Const Wednesday As FirstDayOfWeek

2 [JScript] public var Wednesday : FirstDayOfWeek;

3  
4 *Description*

5 The first day of the week is Wednesday.

6 FontInfo class (System.Web.UI.WebControls)

7 ToString

8  
9  
10 *Description*

11 Encapsulates the font properties of text. This class cannot be inherited.

12 Use the **System.Web.UI.WebControls.FontInfo** class to encapsulate the  
13 font properties of text. You can specify the font name and the font size. You can  
14 also specify whether the style of the font is bold, italic, overlined, strikethrough, or  
15 underlined.

16 Bold

17 ToString

18  
19 [C#] public bool Bold {get; set;}

20 [C++] public: \_\_property bool get\_Bold();public: \_\_property void set\_Bold(bool);

21 [VB] Public Property Bold As Boolean

22 [JScript] public function get Bold() : Boolean;public function set Bold(Boolean);

23  
24 *Description*

25 Gets or sets a value that indicates whether the font is bold.

1 Use the **System.Web.UI.WebControls.FontInfo.Bold** property to specify  
2 or determine whether the font is bold.

3 Italic

4 ToString

5  
6 [C#] public bool Italic {get; set;}

7 [C++] public: \_\_property bool get\_Italic();public: \_\_property void set\_Italic(bool);

8 [VB] Public Property Italic As Boolean

9 [JScript] public function get Italic() : Boolean;public function set Italic(Boolean);

10  
11 *Description*

12 Gets or sets a value that indicates whether the font is italic.

13 Use the **System.Web.UI.WebControls.FontInfo.Italic** property to specify  
14 or determine whether the font is italic.

15 Name

16 ToString

17  
18 [C#] public string Name {get; set;}

19 [C++] public: \_\_property String\* get\_Name();public: \_\_property void  
20 set\_Name(String\*);

21 [VB] Public Property Name As String

22 [JScript] public function get Name() : String;public function set Name(String);

23  
24 *Description*

25 Gets or sets the primary font name.

1 Use the **System.Web.UI.WebControls.FontInfo.Name** property to  
2 specify or determine the primary font name. The primary font name determines  
3 the font that is used to display text in the control associated with the  
4 **System.Web.UI.WebControls.FontInfo** .

5 Names

6 ToString

7  
8 [C#] public string[] Names {get; set;}

9 [C++] public: \_\_property String\* get\_Names();public: \_\_property void  
10 set\_Names(String\* \_\_gc[]);

11 [VB] Public Property Names As String ()

12 [JScript] public function get Names() : String[];public function set  
13 Names(String[]);

14  
15 *Description*

16 Gets or sets an ordered array of font names.

17 Use the **System.Web.UI.WebControls.FontInfo.Names** property to  
18 specify or determine an ordered array of font names for a  
19 **System.Web.UI.WebControls.FontInfo** . The  
20 **System.Web.UI.WebControls.FontInfo.Names** property is commonly used to  
21 store a list of available font names.

22 Overline

23 ToString

24  
25 [C#] public bool Overline {get; set;}

[C++] public: \_\_property bool get\_Overline();public: \_\_property void  
set\_Overline(bool);

[VB] Public Property Overline As Boolean

[JScript] public function get Overline() : Boolean;public function set  
Overline(Boolean);

#### *Description*

Gets or sets a value that indicates whether the font is overlined.

Use the **System.Web.UI.WebControls.FontInfo.Overline** property to  
specify or determine whether the font contains a line above the text.

Size

ToString

[C#] public FontUnit Size {get; set;}

[C++] public: \_\_property FontUnit get\_Size();public: \_\_property void  
set\_Size(FontUnit);

[VB] Public Property Size As FontUnit

[JScript] public function get Size() : FontUnit;public function set Size(FontUnit);

#### *Description*

Gets or sets the font size.

Use the **System.Web.UI.WebControls.FontInfo.Size** property to specify  
the size of the font.

Strikeout

ToString



1  
2 [C#] public bool Strikeout {get; set;}

3 [C++] public: \_\_property bool get\_Strikeout();public: \_\_property void  
4 set\_Strikeout(bool);

5 [VB] Public Property Strikeout As Boolean

6 [JScript] public function get Strikeout() : Boolean;public function set  
7 Strikeout(Boolean);

8  
9 *Description*

10 Gets or sets a value that indicates whether the font is strikethrough.

11 Use the **System.Web.UI.WebControls.FontInfo.Strikeout** property to  
12 specify or determine whether the font contains a line through the text.

13 Underline

14 ToString

15  
16 [C#] public bool Underline {get; set;}

17 [C++] public: \_\_property bool get\_Underline();public: \_\_property void  
18 set\_Underline(bool);

19 [VB] Public Property Underline As Boolean

20 [JScript] public function get Underline() : Boolean;public function set  
21 Underline(Boolean);

22  
23 *Description*

24 Gets or sets a value that indicates whether the font is underlined.

1 Use the **System.Web.UI.WebControls.FontInfo.Underline** property to  
2 specify or determine whether the font contains a line under the text.

### 3 CopyFrom

4  
5 [C#] public void CopyFrom(FontInfo f);  
6 [C++] public: void CopyFrom(FontInfo\* f);  
7 [VB] Public Sub CopyFrom(ByVal f As FontInfo)  
8 [JScript] public function CopyFrom(f : FontInfo);  
9

### 10 *Description*

11 Duplicates the font properties of the specified  
12 **System.Web.UI.WebControls.FontInfo** into the current instance of the  
13 **System.Web.UI.WebControls.FontInfo** class.

14 Use the  
15 **System.Web.UI.WebControls.FontInfo.CopyFrom(System.Web.UI.WebCont**  
16 **rols.FontInfo)** method to duplicate the font properties of the specified  
17 **System.Web.UI.WebControls.FontInfo** into the current instance of the  
18 **System.Web.UI.WebControls.FontInfo** class. A  
19 **System.Web.UI.WebControls.FontInfo** that contains the font properties to  
20 duplicate.

### 21 MergeWith

22  
23 [C#] public void MergeWith(FontInfo f);  
24 [C++] public: void MergeWith(FontInfo\* f);  
25 [VB] Public Sub MergeWith(ByVal f As FontInfo)

1 [JScript] public function MergeWith(f : FontInfo);

2  
3 *Description*

4 Combines the font properties of the specified  
5 **System.Web.UI.WebControls.FontInfo** with the current instance of the  
6 **System.Web.UI.WebControls.FontInfo** class.

7 Use the  
8 **System.Web.UI.WebControls.FontInfo.MergeWith(System.Web.UI.WebCont**  
9 **rols.FontInfo)** method to combine the font properties of the specified  
10 **System.Web.UI.WebControls.FontInfo** with the current instance of the  
11 **System.Web.UI.WebControls.FontInfo** class. A  
12 **System.Web.UI.WebControls.FontInfo** that contains the font properties to  
13 combine.

14 ShouldSerializeNames

15  
16 [C#] public bool ShouldSerializeNames();

17 [C++] public: bool ShouldSerializeNames();

18 [VB] Public Function ShouldSerializeNames() As Boolean

19 [JScript] public function ShouldSerializeNames() : Boolean;

20  
21 *Description*

22 Determines whether the **System.Web.UI.WebControls.FontInfo.Names**  
23 property contains enough entries to serialize.

24 *Return Value:* **true** if the **System.Web.UI.WebControls.FontInfo.Names**  
25 property contains enough entries to serialize; otherwise, **false** .

1        Use the **System.Web.UI.WebControls.FontInfo.ShouldSerializeNames**  
2 method to determine whether the  
3 **System.Web.UI.WebControls.FontInfo.Names** property contains enough entries  
4 to serialize.

5        ToString

6  
7 [C#] public override string ToString();  
8 [C++] public: String\* ToString();  
9 [VB] Overrides Public Function ToString() As String  
10 [JScript] public override function ToString() : String;

11  
12 *Description*

13        Returns a string that contains the font name and size for an instance of the  
14 **System.Web.UI.WebControls.FontInfo** class.

15 *Return Value:* A string that contains the font name and size for an instance of the  
16 **System.Web.UI.WebControls.FontInfo** class.

17        Use the **System.Web.UI.WebControls.FontInfo.ToString** method to  
18 create a string that contains the font name and size, separated by a comma, for an  
19 instance of the **System.Web.UI.WebControls.FontInfo** class.

20        FontNamesConverter class (System.Web.UI.WebControls)

21        ToString

22  
23  
24 *Description*  
25

Converts a string containing a list of font names to an array of strings containing the individual names. It also performs the reverse function.

Use the

**System.Web.UI.WebControls.FontNamesConverter.ConvertFrom(System.ComponentModel.ITypeDescriptorContext, System.Globalization.CultureInfo, System.Object)** method of this class to convert a single string containing a list of font names to an array of strings containing the individual names. Each font name in the string must be separated by a comma. For example, the string, "arial, times new roman, verdana", converts to an array that contains the strings "arial", "times new roman", and "verdana". Notice the commas are removed along with any white space at the beginning or end of the font name. White space in the middle of a font name is not removed.

FontNamesConverter

*Example Syntax:*

ToString

[C#] public FontNamesConverter();

[C++] public: FontNamesConverter();

[VB] Public Sub New()

[JScript] public function FontNamesConverter();

CanConvertFrom

[C#] public override bool CanConvertFrom(ITypeDescriptorContext context, Type sourceType);

[C++] public: bool CanConvertFrom(ITypeDescriptorContext\* context, Type\*

1 sourceType);

2 [VB] Overrides Public Function CanConvertFrom(ByVal context As

3 ITypeDescriptorContext, ByVal sourceType As Type) As Boolean

4 [JScript] public override function CanConvertFrom(context :

5 ITypeDescriptorContext, sourceType : Type) : Boolean;

### 7 *Description*

8 Determines whether this converter can convert an object of the specified  
9 data type to an array of strings containing individual font names.

10 *Return Value:* **true** if the type can be converted; otherwise, **false** .

11 Use the

12 **System.Web.UI.WebControls.FontNamesConverter.CanConvertFrom(System**  
13 **ComponentModel.ITypeDescriptorContext,System.Type)** method to  
14 determine whether the specified data type can be converted to an array of strings  
15 containing individual font names. A

16 **System.ComponentModel.ITypeDescriptorContext** that provides information  
17 about the context of a type converter. You can optionally pass in **null** for this  
18 parameter. A **System.Type** that represents the data type to convert from. This  
19 parameter must be set to the **string** data type.

### 20 *ConvertFrom*

21  
22 [C#] public override object ConvertFrom(ITypeDescriptorContext context,  
23 CultureInfo culture, object value);

24 [C++] public: Object\* ConvertFrom(ITypeDescriptorContext\* context,  
25 CultureInfo\* culture, Object\* value);

```

1 [VB] Overrides Public Function ConvertFrom(ByVal context As
2 ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object)
3 As Object
4 [JScript] public override function ConvertFrom(context : ITypeDescriptorContext,
5 culture : CultureInfo, value : Object) : Object;
6

```

### *Description*

Converts a string that represents a list of font names into an array of strings containing individual font names.

**Return Value:** A **System.Object** that represents the array of strings containing the individual font names.

Use the **System.Web.UI.WebControls.FontNamesConverter.ConvertFrom(System.ComponentModel.ITypeDescriptorContext, System.Globalization.CultureInfo, System.Object)** method to convert a single **string** containing a list of font names to an array of strings containing the individual names. Each font name in the string must be separated by a comma. For example, the string, "arial, times new roman, verdana", converts to an array that contains the strings "arial", "times new roman", and "verdana". Notice that the commas are removed along with any white space at the beginning or end of the font name. White space in the middle of a font name is not removed. A **System.ComponentModel.ITypeDescriptorContext** that provides information about the context of a type converter. This parameter is not used in this method. It is reserved for future versions of this method. You can optionally pass in **null** for this parameter. A **System.Globalization.CultureInfo** object that represents information about a culture such as language, calendar

1 system, and so on. This parameter is not used in this method. It is reserved for  
2 future versions of this method. You can optionally pass in **null** for this parameter.

3 A **System.Object** that represents the source string to convert from.

#### 4 ConvertTo

5  
6 [C#] public override object ConvertTo(ITypeDescriptorContext context,  
7 CultureInfo culture, object value, Type destinationType);

8 [C++] public: Object\* ConvertTo(ITypeDescriptorContext\* context, CultureInfo\*  
9 culture, Object\* value, Type\* destinationType);

10 [VB] Overrides Public Function ConvertTo(ByVal context As  
11 ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object,  
12 ByVal destinationType As Type) As Object

13 [JScript] public override function ConvertTo(context : ITypeDescriptorContext,  
14 culture : CultureInfo, value : Object, destinationType : Type) : Object;

#### 16 Description

17 Creates a string that represents a list of font names from an array of strings  
18 containing individual font names.

19 *Return Value:* A **System.Object** that represents a string containing a list of font  
20 names.

21 Use the

22 **System.Web.UI.WebControls.FontNamesConverter.ConvertTo(System.Com**  
23 **ponentModel.ITypeDescriptorContext,System.Globalization.CultureInfo,Syst**  
24 **em.Object,System.Type)** method to convert an array of strings containing the  
25 individual font names to a single string containing a list of the names. For



example, an array that contains the strings "arial", "times new roman", and "verdana" converts to the string, "arial,times new roman,verdana". Notice that commas are automatically inserted between the font names without any white space. A **System.ComponentModel.ITypeDescriptorContext** object that provides information about the context of a type converter. This parameter is not used in this method. It is reserved for future versions of this method. You can optionally pass in **null** for this parameter. A **System.Globalization.CultureInfo** object that represents information about a culture such as language, calendar system, and so on. This parameter is not used in this method. It is reserved for future versions of this method. You can optionally pass in **null** for this parameter. A **System.Object** that represents the source array of strings to convert from. A **System.Type** object that represents the data type to convert to. This parameter must be set to the **string** data type.

FontSize enumeration (System.Web.UI.WebControls)

ToString

### *Description*

Specifies the font sizes defined by HTML 4.0.

The **System.Web.UI.WebControls.FontSize** enumeration represents the font sizes defined by HTML 4.0. The font size can be an absolute or relative size.

ToString

[C#] public const FontSize AsUnit;

[C++] public: const FontSize AsUnit;

1 [VB] Public Const AsUnit As FontSize

2 [JScript] public var AsUnit : FontSize;

3

4 *Description*

5 The font size is specified by a point value.

6 ToString

7

8 [C#] public const FontSize Large;

9 [C++] public: const FontSize Large;

10 [VB] Public Const Large As FontSize

11 [JScript] public var Large : FontSize;

12

13 *Description*

14 The font size is two sizes larger than the base font size.

15 ToString

16

17 [C#] public const FontSize Larger;

18 [C++] public: const FontSize Larger;

19 [VB] Public Const Larger As FontSize

20 [JScript] public var Larger : FontSize;

21

22 *Description*

23 The font size is one size larger than the parent element.

24 ToString

25

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#] public const FontSize Medium;  
[C++] public: const FontSize Medium;  
[VB] Public Const Medium As FontSize  
[JScript] public var Medium : FontSize;
```

*Description*

The font size is one size larger than the default font size.  
ToString

```
[C#] public const FontSize NotSet;  
[C++] public: const FontSize NotSet;  
[VB] Public Const NotSet As FontSize  
[JScript] public var NotSet : FontSize;
```

*Description*

The font size is not set.  
ToString

```
[C#] public const FontSize Small;  
[C++] public: const FontSize Small;  
[VB] Public Const Small As FontSize  
[JScript] public var Small : FontSize;
```

*Description*

The base font size determined by the browser.

ToString

[C#] public const FontSize Smaller;

[C++] public: const FontSize Smaller;

[VB] Public Const Smaller As FontSize

[JScript] public var Smaller : FontSize;

*Description*

The font size is one size smaller than the parent element.

ToString

[C#] public const FontSize XLarge;

[C++] public: const FontSize XLarge;

[VB] Public Const XLarge As FontSize

[JScript] public var XLarge : FontSize;

*Description*

The font size is three sizes larger than the base font size.

ToString

[C#] public const FontSize XSmall;

[C++] public: const FontSize XSmall;

[VB] Public Const XSmall As FontSize

[JScript] public var XSmall : FontSize;

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

*Description*

The font size is one size smaller than the base font size.

ToString

[C#] public const FontSize XXLarge;

[C++] public: const FontSize XXLarge;

[VB] Public Const XXLarge As FontSize

[JScript] public var XXLarge : FontSize;

*Description*

The font size is four sizes larger than the base font size.

ToString

[C#] public const FontSize XXSmall;

[C++] public: const FontSize XXSmall;

[VB] Public Const XXSmall As FontSize

[JScript] public var XXSmall : FontSize;

*Description*

The font size is two sizes smaller than the base font size.

FontUnit structure (System.Web.UI.WebControls)

ToString

1  
2  
3 *Description*

4 Represents the size of a font.

5 For a list of initial property values for an instance of  
6 **System.Web.UI.WebControls.FontUnit**, see the  
7 **System.Web.UI.WebControls.FontUnit.#ctor** constructor.

8 ToString

9  
10 [C#] public static readonly FontUnit Empty;

11 [C++] public: static FontUnit Empty;

12 [VB] Public Shared ReadOnly Empty As FontUnit

13 [JScript] public static var Empty : FontUnit;

14  
15 *Description*

16 Represents an empty **System.Web.UI.WebControls.FontUnit** object. This  
17 field is read only.

18 Use the **System.Web.UI.WebControls.FontUnit.Empty** field to represent  
19 an empty **System.Web.UI.WebControls.FontUnit** object. No HTML rendered  
20 output is created when this field is used.

21 ToString

22  
23 [C#] public static readonly FontUnit Large;

24 [C++] public: static FontUnit Large;

25 [VB] Public Shared ReadOnly Large As FontUnit

1 [JScript] public static var Large : FontUnit;

3 *Description*

4 Represents a **System.Web.UI.WebControls.FontUnit** object with the  
5 **System.Web.UI.WebControls.FontUnit.Type** property set to **FontSize.Large**  
6 font. This field is read only.

7 Use the **System.Web.UI.WebControls.FontUnit.Large** field to represent  
8 a **System.Web.UI.WebControls.FontUnit** object with the  
9 **System.Web.UI.WebControls.FontUnit.Type** property set to **FontSize.Large** .

10 ToString

12 [C#] public static readonly FontUnit Larger;

13 [C++] public: static FontUnit Larger;

14 [VB] Public Shared ReadOnly Larger As FontUnit

15 [JScript] public static var Larger : FontUnit;

17 *Description*

18 Represents a **System.Web.UI.WebControls.FontUnit** object with the  
19 **System.Web.UI.WebControls.FontUnit.Type** property set to **FontSize.Larger**  
20 font. This field is read only.

21 Use the **System.Web.UI.WebControls.FontUnit.Larger** field to represent  
22 a **System.Web.UI.WebControls.FontUnit** object with the  
23 **System.Web.UI.WebControls.FontUnit.Type** property set to **FontSize.Larger** .

24 ToString

```

1
2 [C#] public static readonly FontUnit Medium;
3 [C++] public: static FontUnit Medium;
4 [VB] Public Shared ReadOnly Medium As FontUnit
5 [JScript] public static var Medium : FontUnit;
6

```

### 7 *Description*

8 Represents a **System.Web.UI.WebControls.FontUnit** object with the  
9 **System.Web.UI.WebControls.FontUnit.Type** property set to **FontSize.Medium**  
10 font. This field is read only.

11 Use the **System.Web.UI.WebControls.FontUnit.Medium** field to  
12 represent a **System.Web.UI.WebControls.FontUnit** object with the  
13 **System.Web.UI.WebControls.FontUnit.Type** property set to **FontSize.Medium**

### 14 15 ToString

```

16
17 [C#] public static readonly FontUnit Small;
18 [C++] public: static FontUnit Small;
19 [VB] Public Shared ReadOnly Small As FontUnit
20 [JScript] public static var Small : FontUnit;
21

```

### 22 *Description*

23 Represents a **System.Web.UI.WebControls.FontUnit** object with the  
24 **System.Web.UI.WebControls.FontUnit.Type** property set to **FontSize.Small**  
25 font. This field is read only.



1 Use the **System.Web.UI.WebControls.FontUnit.Small** field to represent  
2 a **System.Web.UI.WebControls.FontUnit** object with the  
3 **System.Web.UI.WebControls.FontUnit.Type** property set to **FontSize.Small** .

4 ToString

5  
6 [C#] public static readonly FontUnit Smaller;

7 [C++] public: static FontUnit Smaller;

8 [VB] Public Shared ReadOnly Smaller As FontUnit

9 [JScript] public static var Smaller : FontUnit;

10  
11 *Description*

12 Represents a **System.Web.UI.WebControls.FontUnit** object with the  
13 **System.Web.UI.WebControls.FontUnit.Type** property set to **FontSize.Smaller**  
14 font. This field is read only.

15 Use the **System.Web.UI.WebControls.FontUnit.Smaller** field to  
16 represent a **System.Web.UI.WebControls.FontUnit** object with the  
17 **System.Web.UI.WebControls.FontUnit.Type** property set to **FontSize.Smaller**

18  
19 ToString

20  
21 [C#] public static readonly FontUnit XLarge;

22 [C++] public: static FontUnit XLarge;

23 [VB] Public Shared ReadOnly XLarge As FontUnit

24 [JScript] public static var XLarge : FontUnit;

1  
2 *Description*

3 Represents a **System.Web.UI.WebControls.FontUnit** object with the  
4 **System.Web.UI.WebControls.FontUnit.Type** property set to **FontSize.XLarge**  
5 font. This field is read only.

6 Use the **System.Web.UI.WebControls.FontUnit.XLarge** field to  
7 represent a **System.Web.UI.WebControls.FontUnit** object with the  
8 **System.Web.UI.WebControls.FontUnit.Type** property set to **FontSize.XLarge** .

9 ToString

10  
11 [C#] public static readonly FontUnit XSmall;

12 [C++] public: static FontUnit XSmall;

13 [VB] Public Shared ReadOnly XSmall As FontUnit

14 [JScript] public static var XSmall : FontUnit;

15  
16 *Description*

17 Represents a **System.Web.UI.WebControls.FontUnit** object with the  
18 **System.Web.UI.WebControls.FontUnit.Type** property set to **FontSize.XSmall**  
19 font. This field is read only.

20 Use the **System.Web.UI.WebControls.FontUnit.XSmall** field to  
21 represent a **System.Web.UI.WebControls.FontUnit** object with the  
22 **System.Web.UI.WebControls.FontUnit.Type** property set to **FontSize.XSmall** .

23 ToString

24  
25 [C#] public static readonly FontUnit XXLarge;

1 [C++] public: static FontUnit XXLarge;

2 [VB] Public Shared ReadOnly XXLarge As FontUnit

3 [JScript] public static var XXLarge : FontUnit;

4  
5 *Description*

6 Represents a **System.Web.UI.WebControls.FontUnit** object with the  
7 **System.Web.UI.WebControls.FontUnit.Type** property set to  
8 **FontSize.XXLarge** font. This field is read only.

9 Use the **System.Web.UI.WebControls.FontUnit.XXLarge** field to  
10 represent a **System.Web.UI.WebControls.FontUnit** object with the  
11 **System.Web.UI.WebControls.FontUnit.Type** property set to  
12 **FontSize.XXLarge** .

13 ToString

14  
15 [C#] public static readonly FontUnit XXSmall;

16 [C++] public: static FontUnit XXSmall;

17 [VB] Public Shared ReadOnly XXSmall As FontUnit

18 [JScript] public static var XXSmall : FontUnit;

19  
20 *Description*

21 Represents a **System.Web.UI.WebControls.FontUnit** object with the  
22 **System.Web.UI.WebControls.FontUnit.Type** property set to  
23 **FontSize.XXSmall** font. This field is read only.

24 Use the **System.Web.UI.WebControls.FontUnit.XXSmall** field to  
25 represent a **System.Web.UI.WebControls.FontUnit** object with the

1 **System.Web.UI.WebControls.FontUnit.Type** property set to

2 **FontSize.XXSmall** .

3       FontUnit

4       *Example Syntax:*

5       ToString

6  
7 [C#] public FontUnit(FontSize type);

8 [C++] public: FontUnit(FontSize type);

9 [VB] Public Sub New(ByVal type As FontSize)

10 [JScript] public function FontUnit(type : FontSize); Initializes a new instance of  
11 the **System.Web.UI.WebControls.FontUnit** class.

12  
13 *Description*

14       Initializes a new instance of the **System.Web.UI.WebControls.FontUnit**  
15 class with the specified **System.Web.UI.WebControls.FontSize** object.

16       Use this constructor to create and initialize an instance of the  
17 **System.Web.UI.WebControls.FontUnit** class using the specified  
18 **System.Web.UI.WebControls.FontSize** object. One of the  
19 **System.Web.UI.WebControls.FontSize** values.

20       FontUnit

21       *Example Syntax:*

22       ToString

23  
24 [C#] public FontUnit(int value);

25 [C++] public: FontUnit(int value);

1 [VB] Public Sub New(ByVal value As Integer)

2 [JScript] public function FontUnit(value : int);

3  
4 *Description*

5       Initializes a new instance of the **System.Web.UI.WebControls.FontUnit**  
6 class with the specified font size.

7       Use this constructor to create and initialize an instance of the  
8 **System.Web.UI.WebControls.FontUnit** class using the specified font size. An  
9 integer that represents the size of the font.

10       FontUnit

11       *Example Syntax:*

12       ToString

13  
14 [C#] public FontUnit(string value);

15 [C++] public: FontUnit(String\* value);

16 [VB] Public Sub New(ByVal value As String)

17 [JScript] public function FontUnit(value : String);

18  
19 *Description*

20       Initializes a new instance of the **System.Web.UI.WebControls.FontUnit**  
21 class with the specified font value.

22       Use this constructor to create and initialize an instance of the  
23 **System.Web.UI.WebControls.FontUnit** class using the specified font value. A  
24 string representation of one of the **System.Web.UI.WebControls.FontSize** value.

25       FontUnit

*Example Syntax:*

ToString

```
[C#] public FontUnit(Unit value);  
[C++] public: FontUnit(Unit value);  
[VB] Public Sub New(ByVal value As Unit)  
[JScript] public function FontUnit(value : Unit);
```

*Description*

Initializes a new instance of the **System.Web.UI.WebControls.FontUnit** class with the specified **System.Web.UI.WebControls.Unit** object.

Use this constructor to create and initialize an instance of the **System.Web.UI.WebControls.FontUnit** class using the specified **System.Web.UI.WebControls.Unit** object. A **System.Web.UI.WebControls.Unit** object that specifies the font size.

FontUnit

*Example Syntax:*

ToString

```
[C#] public FontUnit(string value, CultureInfo culture);  
[C++] public: FontUnit(String* value, CultureInfo* culture);  
[VB] Public Sub New(ByVal value As String, ByVal culture As CultureInfo)  
[JScript] public function FontUnit(value : String, culture : CultureInfo);
```

*Description*

1        Initializes a new instance of the **System.Web.UI.WebControls.FontUnit**  
2 class with the specified font value and **System.Globalization.CultureInfo** object.

3        Use this constructor to create and initialize an instance of the  
4 **System.Web.UI.WebControls.FontUnit** class using the specified font value and  
5 **System.Globalization.CultureInfo** object. A string representation of one of the  
6 **System.Web.UI.WebControls.FontSize** value. A  
7 **System.Globalization.CultureInfo** object that represents the culture to create the  
8 **System.Web.UI.WebControls.FontUnit** object for.

9        **IsEmpty**

10       **ToString**

11  
12 [C#] public bool IsEmpty {get;}

13 [C++] public: \_\_property bool get\_IsEmpty();

14 [VB] Public ReadOnly Property IsEmpty As Boolean

15 [JScript] public function get IsEmpty() : Boolean;

16  
17 *Description*

18        Gets a value that indicates whether the font size has been set.

19        The **System.Web.UI.WebControls.FontUnit.IsEmpty** property is  
20 commonly used to determine whether the font size has been set.

21        **Type**

22        **ToString**

23  
24 [C#] public FontSize Type {get;}

25 [C++] public: \_\_property FontSize get\_Type();

1 [VB] Public ReadOnly Property Type As FontSize

2 [JScript] public function get Type() : FontSize;

3  
4 *Description*

5 Gets a **System.Web.UI.WebControls.FontSize** object that represents the  
6 font size.

7 Use the **System.Web.UI.WebControls.FontUnit.Type** property to  
8 determine the font size.

9 Unit

10 ToString

11  
12 [C#] public Unit Unit {get;}

13 [C++] public: \_\_property Unit get \_Unit();

14 [VB] Public ReadOnly Property Unit As Unit

15 [JScript] public function get Unit() : Unit;

16  
17 *Description*

18 Gets a **System.Web.UI.WebControls.Unit** object that represents the font  
19 size.

20 Equals

21  
22 [C#] public override bool Equals(object obj);

23 [C++] public: bool Equals(Object\* obj);

24 [VB] Overrides Public Function Equals(ByVal obj As Object) As Boolean

25 [JScript] public override function Equals(obj : Object) : Boolean;



1  
2 *Description*

3 Determines whether the specified **System.Object** is equivalent to this  
4 instance of the **System.Web.UI.WebControls.FontUnit** class.

5 *Return Value:* **true** if the specified **System.Object** is equivalent to this instance of  
6 the **System.Web.UI.WebControls.FontUnit** class; otherwise, **false** . A

7 **System.Object** that contains the object to compare to this instance.

8 GetHashCode

9  
10 [C#] public override int GetHashCode();

11 [C++] public: int GetHashCode();

12 [VB] Overrides Public Function GetHashCode() As Integer

13 [JScript] public override function GetHashCode() : int;

14  
15 *Description*

16 Returns the hash code for this instance.

17 *Return Value:* A 32-bit signed integer hash code.

18 op\_Equality

19  
20 [C#] public static bool operator ==(FontUnit left, FontUnit right);

21 [C++] public: static bool op\_Equality(FontUnit left, FontUnit right);

22 [VB] returnValue = FontUnit.op\_Equality(left, right)

23 [JScript] returnValue = left == right;

24  
25 *Description*

1 Compares two **System.Web.UI.WebControls.FontUnit** objects for  
2 equality.

3 *Return Value:* **true** if both **System.Web.UI.WebControls.FontUnit** objects are  
4 equal; otherwise, **false** . A **System.Web.UI.WebControls.FontUnit** object that  
5 contains font properties on the left of the operator. A

6 **System.Web.UI.WebControls.FontUnit** object that contains font properties on  
7 the right of the operator.

8 **op\_Implicit**

9  
10 [C#] public static implicit operator FontUnit(int n);

11 [C++] public: static FontUnit op\_Implicit(int n);

12 [VB] returnValue = FontUnit.op\_Implicit(n)

13 [JScript] returnValue = n;

14  
15 *Description*

16 Implicitly creates a **System.Web.UI.WebControls.FontUnit** of type  
17 **System.Drawing.Point** from an integer value. An integer representing the  
18 **System.Drawing.Point** value to convert into a  
19 **System.Web.UI.WebControls.FontUnit**.

20 **op\_Inequality**

21  
22 [C#] public static bool operator !=(FontUnit left, FontUnit right);

23 [C++] public: static bool op\_Inequality(FontUnit left, FontUnit right);

24 [VB] returnValue = FontUnit.op\_Inequality(left, right)

25 [JScript] returnValue = left != right;

1  
2 *Description*

3 Compares two **System.Web.UI.WebControls.FontUnit** objects for  
4 inequality.

5 *Return Value:* **true** if both **System.Web.UI.WebControls.FontUnit** objects are  
6 not equal; otherwise, **false** . A **System.Web.UI.WebControls.FontUnit** that  
7 contains font properties on the left of the operator. A  
8 **System.Web.UI.WebControls.FontUnit** that contains font properties on the right  
9 of the operator.

10 Parse

11  
12 [C#] public static FontUnit Parse(string s);

13 [C++] public: static FontUnit Parse(String\* s);

14 [VB] Public Shared Function Parse(ByVal s As String) As FontUnit

15 [JScript] public static function Parse(s : String) : FontUnit; Converts the specified  
16 string to its **System.Web.UI.WebControls.FontUnit** equivalent.

17  
18 *Description*

19 Converts the specified string to its default  
20 **System.Web.UI.WebControls.FontUnit** equivalent. A string representation of  
21 one of the **System.Web.UI.WebControls.FontSize** value.

22 Parse

23  
24 [C#] public static FontUnit Parse(string s, CultureInfo culture);

25 [C++] public: static FontUnit Parse(String\* s, CultureInfo\* culture);

1 [VB] Public Shared Function Parse(ByVal s As String, ByVal culture As  
2 CultureInfo) As FontUnit  
3 [JScript] public static function Parse(s : String, culture : CultureInfo) : FontUnit;  
4

5 *Description*

6 Converts the specified string to its  
7 **System.Web.UI.WebControls.FontUnit** equivalent in the specified culture. A  
8 string representation of one of the **System.Web.UI.WebControls.FontSize** value.  
9 A **System.Globalization.CultureInfo** object that represents the culture of the  
10 **System.Web.UI.WebControls.FontUnit** object.

11 Point

12  
13 [C#] public static FontUnit Point(int n);  
14 [C++] public: static FontUnit Point(int n);  
15 [VB] Public Shared Function Point(ByVal n As Integer) As FontUnit  
16 [JScript] public static function Point(n : int) : FontUnit;  
17

18 *Description*

19 Creates a **System.Web.UI.WebControls.FontUnit** of type  
20 **System.Drawing.Point** from an integer value.  
21 *Return Value:* A **System.Web.UI.WebControls.FontUnit** that represents the new  
22 font size. An integer representing the **System.Drawing.Point** value to convert into  
23 a **System.Web.UI.WebControls.FontUnit**.

24 ToString

```

1
2 [C#] public override string ToString();
3 [C++] public: String* ToString();
4 [VB] Overrides Public Function ToString() As String
5 [JScript] public override function ToString() : String; Converts the
6 System.Web.UI.WebControls.FontUnit object to a string representation.

```

#### *Description*

Converts the **System.Web.UI.WebControls.FontUnit** object to the default string representation.

*Return Value:* A string representation of the **System.Web.UI.WebControls.FontUnit** object.

#### **ToString**

```

15 [C#] public string ToString(CultureInfo culture);
16 [C++] public: String* ToString(CultureInfo* culture);
17 [VB] Public Function ToString(ByVal culture As CultureInfo) As String
18 [JScript] public function ToString(culture : CultureInfo) : String;

```

#### *Description*

Converts the **System.Web.UI.WebControls.FontUnit** object to a string representation using the specified **System.Globalization.CultureInfo** object. A **System.Globalization.CultureInfo** object that contains the culture.

FontUnitConverter class (System.Web.UI.WebControls)

#### **ToString**

1  
2  
3 *Description*

4        Converts a **System.Web.UI.WebControls.FontUnit** to and from a  
5 specified data type.

6        FontUnitConverter

7        *Example Syntax:*

8        ToString

9  
10 [C#] public FontUnitConverter();

11 [C++] public: FontUnitConverter();

12 [VB] Public Sub New()

13 [JScript] public function FontUnitConverter();

14        CanConvertFrom

15  
16 [C#] public override bool CanConvertFrom(ITypeDescriptorContext context,  
17 Type sourceType);

18 [C++] public: bool CanConvertFrom(ITypeDescriptorContext\* context, Type\*  
19 sourceType);

20 [VB] Overrides Public Function CanConvertFrom(ByVal context As  
21 ITypeDescriptorContext, ByVal sourceType As Type) As Boolean

22 [JScript] public override function CanConvertFrom(context :  
23 ITypeDescriptorContext, sourceType : Type) : Boolean;

24  
25 *Description*

Determines if the specified data type can be converted to a

**System.Web.UI.WebControls.FontUnit** .

*Return Value:* **true** if the type can be converted; otherwise **false** . An

**System.ComponentModel.ITypeDescriptorContext** that provides information about the context of a type converter. A **System.Type** that represents the data type to check.

**ConvertFrom**

[C#] public override object ConvertFrom(ITypeDescriptorContext context, CultureInfo culture, object value);

[C++] public: Object\* ConvertFrom(ITypeDescriptorContext\* context, CultureInfo\* culture, Object\* value);

[VB] Overrides Public Function ConvertFrom(ByVal context As ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object) As Object

[JScript] public override function ConvertFrom(context : ITypeDescriptorContext, culture : CultureInfo, value : Object) : Object;

### *Description*

Converts the specified **System.Object** into a

**System.Web.UI.WebControls.FontUnit** . An

**System.ComponentModel.ITypeDescriptorContext** that provides information about the context of a type converter. A **System.Globalization.CultureInfo** object that represents information about a culture such as language, calendar system, and so on. This parameter is not used in this method. It is reserved for

1 future versions of this method. You can optionally pass in **null** for this parameter.

2 The **System.Object** to convert to a **System.Web.UI.WebControls.FontUnit**.

### 3 ConvertTo

4  
5 [C#] public override object ConvertTo(ITypeDescriptorContext context,  
6 CultureInfo culture, object value, Type destinationType);

7 [C++] public: Object\* ConvertTo(ITypeDescriptorContext\* context, CultureInfo\*  
8 culture, Object\* value, Type\* destinationType);

9 [VB] Overrides Public Function ConvertTo(ByVal context As  
10 ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object,  
11 ByVal destinationType As Type) As Object

12 [JScript] public override function ConvertTo(context : ITypeDescriptorContext,  
13 culture : CultureInfo, value : Object, destinationType : Type) : Object;

### 14 Description

15  
16 Converts the specified **System.Web.UI.WebControls.FontUnit** into the  
17 specified **System.Type**. An

18 **System.ComponentModel.ITypeDescriptorContext** that provides information  
19 about the context of a type converter. A **System.Globalization.CultureInfo**

20 object that represents information about a culture such as language, calendar  
21 system, and so on. This parameter is not used in this method. It is reserved for  
22 future versions of this method. You can optionally pass in **null** for this parameter.

23 A **System.Object** that represents the source array of strings to convert from. A

24 **System.Type** that represents the data type to convert to.

### 25 GetStandardValues



```

1  [C#] public override StandardValuesCollection
2
3  GetStandardValues(ITypeDescriptorContext context);
4
5  [C++] public: StandardValuesCollection*
6
7  GetStandardValues(ITypeDescriptorContext* context);
8
9  [VB] Overrides Public Function GetStandardValues(ByVal context As
10 ITypeDescriptorContext) As StandardValuesCollection
11
12 [JScript] public override function GetStandardValues(context :
13 ITypeDescriptorContext) : StandardValuesCollection;

```

#### *Description*

Returns a

**System.ComponentModel.TypeConverter.StandardValuesCollection**  
containing standard **System.Web.UI.WebControls.FontUnit** values.

*Return Value:* A

**System.ComponentModel.TypeConverter.StandardValuesCollection**  
containing **System.Web.UI.WebControls.FontUnit** values. An  
**System.ComponentModel.ITypeDescriptorContext** that provides information  
about the context of a type converter.

**GetStandardValuesExclusive**

```

21
22 [C#] public override bool GetStandardValuesExclusive(ITypeDescriptorContext
23 context);
24
25 [C++] public: bool GetStandardValuesExclusive(ITypeDescriptorContext*
context);

```

1 [VB] Overrides Public Function GetStandardValuesExclusive(ByVal context As  
2 ITypeDescriptorContext) As Boolean

3 [JScript] public override function GetStandardValuesExclusive(context :  
4 ITypeDescriptorContext) : Boolean;

5  
6 *Description*

7 Indicates whether the specified context contains exclusive standard values.

8 *Return Value:* **false** for all cases. An

9 **System.ComponentModel.ITypeDescriptorContext** that provides information  
10 about the context of a type converter.

11 GetStandardValuesSupported

12  
13 [C#] public override bool GetStandardValuesSupported(ITypeDescriptorContext  
14 context);

15 [C++] public: bool GetStandardValuesSupported(ITypeDescriptorContext\*  
16 context);

17 [VB] Overrides Public Function GetStandardValuesSupported(ByVal context As  
18 ITypeDescriptorContext) As Boolean

19 [JScript] public override function GetStandardValuesSupported(context :  
20 ITypeDescriptorContext) : Boolean;

21  
22 *Description*

23 Indicates whether the specified context contains supported standard values.

24 *Return Value:* **true** for all cases. An

1 **System.ComponentModel.ITypeDescriptorContext** that provides information  
2 about the context of a type converter.

3 GridLines enumeration (System.Web.UI.WebControls)

4 ToString

5  
6  
7 *Description*

8 Specifies the grid line styles for controls displaying items in a table  
9 structure.

10 The **System.Web.UI.WebControls.GridLines** enumeration represents the  
11 grid line styles for controls displaying items in a table structure.

12 ToString

13  
14 [C#] public const GridLines Both;

15 [C++] public: const GridLines Both;

16 [VB] Public Const Both As GridLines

17 [JScript] public var Both : GridLines;

18  
19 *Description*

20 Both horizontal and vertical grid lines rendered.

21 ToString

22  
23 [C#] public const GridLines Horizontal;

24 [C++] public: const GridLines Horizontal;

25 [VB] Public Const Horizontal As GridLines

1 [JScript] public var Horizontal : GridLines;

3 *Description*

4 Only horizontal grid lines rendered.

5 ToString

7 [C#] public const GridLines None;

8 [C++] public: const GridLines None;

9 [VB] Public Const None As GridLines

10 [JScript] public var None : GridLines;

12 *Description*

13 No grid lines rendered.

14 ToString

16 [C#] public const GridLines Vertical;

17 [C++] public: const GridLines Vertical;

18 [VB] Public Const Vertical As GridLines

19 [JScript] public var Vertical : GridLines;

21 *Description*

22 Only vertical grid lines rendered.

23 HorizontalAlign enumeration (System.Web.UI.WebControls)

24 ToString

1  
2  
3 *Description*

4 Specifies the horizontal alignment of items within a container.

5 The **System.Web.UI.WebControls.HorizontalAlign** enumeration  
6 represents the horizontal alignment options for items within a container, such as a  
7 **System.Web.UI.WebControls.TableCell** .

8 ToString

9  
10 [C#] public const HorizontalAlign Center;

11 [C++] public: const HorizontalAlign Center;

12 [VB] Public Const Center As HorizontalAlign

13 [JScript] public var Center : HorizontalAlign;

14  
15 *Description*

16 The contents of a container are centered.

17 ToString

18  
19 [C#] public const HorizontalAlign Justify;

20 [C++] public: const HorizontalAlign Justify;

21 [VB] Public Const Justify As HorizontalAlign

22 [JScript] public var Justify : HorizontalAlign;

23  
24 *Description*

1 The contents of a container are uniformly spread out and aligned with both  
2 the left and right margins.

3 ToString

4  
5 [C#] public const HorizontalAlign Left;

6 [C++] public: const HorizontalAlign Left;

7 [VB] Public Const Left As HorizontalAlign

8 [JScript] public var Left : HorizontalAlign;

9  
10 *Description*

11 The contents of a container are left justified.

12 ToString

13  
14 [C#] public const HorizontalAlign NotSet;

15 [C++] public: const HorizontalAlign NotSet;

16 [VB] Public Const NotSet As HorizontalAlign

17 [JScript] public var NotSet : HorizontalAlign;

18  
19 *Description*

20 The horizontal alignment is not set.

21 ToString

22  
23 [C#] public const HorizontalAlign Right;

24 [C++] public: const HorizontalAlign Right;

25 [VB] Public Const Right As HorizontalAlign

1 [JScript] public var Right : HorizontalAlign;

2  
3 *Description*

4 The contents of a container are right justified.

5 HyperLink class (System.Web.UI.WebControls)

6 ToString

7  
8  
9 *Description*

10 A control that displays a link to another Web page.

11 Use the **System.Web.UI.WebControls.HyperLink** control to create a link  
12 to another Web page. The **System.Web.UI.WebControls.HyperLink** control is  
13 typically displayed as text specified by the

14 **System.Web.UI.WebControls.HyperLink.Text** property. It can also be  
15 displayed as an image specified by the

16 **System.Web.UI.WebControls.HyperLink.ImageUrl** property.

17 HyperLink

18 *Example Syntax:*

19 ToString

20  
21 [C#] public HyperLink();

22 [C++] public: HyperLink();

23 [VB] Public Sub New()

24 [JScript] public function HyperLink();

1  
2 *Description*

3       Initializes a new instance of the  
4 **System.Web.UI.WebControls.HyperLink** class.

5       Use this constructor to create and initialize a new instance of the  
6 **System.Web.UI.WebControls.HyperLink** control.

7       AccessKey

8       Attributes

9       BackColor

10       BorderColor

11       BorderStyle

12       BorderWidth

13       ChildControlsCreated

14       ClientID

15       Context

16       Controls

17       ControlStyle

18       ControlStyleCreated

19       CssClass

20       Enabled

21       EnableViewState

22       Events

23       Font

24       ForeColor

25       HasChildViewState



Height

ID

ImageUrl

ToString

*Description*

Gets or sets the path to an image to display for the **System.Web.UI.WebControls.HyperLink** control.

The **System.Web.UI.WebControls.HyperLink** control can be displayed as text or an image. Use the **System.Web.UI.WebControls.HyperLink.ImageUrl** property to specify an image to display for the **System.Web.UI.WebControls.HyperLink** control.

IsTrackingViewState

NamingContainer

NavigateUrl

ToString

*Description*

Gets or sets the URL to link to when the **System.Web.UI.WebControls.HyperLink** control is clicked.

Use the **System.Web.UI.WebControls.HyperLink.NavigateUrl** property to specify the URL to navigate to when the **System.Web.UI.WebControls.HyperLink** control is clicked.

1 Page  
2 Parent  
3 Site  
4 Style  
5 TabIndex  
6 TagKey  
7 TagName  
8 Target  
9 ToString

*Description*

Gets or sets the target window or frame to display the Web page content linked to when the **System.Web.UI.WebControls.HyperLink** control is clicked.

Use the **System.Web.UI.WebControls.HyperLink.Target** property to specify the frame or window that displays the Web page linked to when the **System.Web.UI.WebControls.HyperLink** control is clicked. The Web page is specified by setting the **System.Web.UI.WebControls.HyperLink.NavigateUrl** property.

20 TemplateSourceDirectory  
21 Text  
22 ToString

*Description*

Gets or sets the text caption for the  
**System.Web.UI.WebControls.HyperLink** control.

The **System.Web.UI.WebControls.HyperLink** control can be displayed as text or an image. Use the **System.Web.UI.WebControls.HyperLink.Text** property to specify the text to display for the **System.Web.UI.WebControls.HyperLink** control.

ToolTip

UniqueID

ViewState

ViewStateIgnoresCase

Visible

Width

AddAttributesToRender

[C#] protected override void AddAttributesToRender(HtmlTextWriter writer);

[C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);

[VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As  
HtmlTextWriter)

[JScript] protected override function AddAttributesToRender(writer :  
HtmlTextWriter);

### *Description*

Adds the attributes of the **System.Web.UI.WebControls.HyperLink** to the output stream for rendering. The output stream to render on the client.

AddParsedSubObject

1  
2 [C#] protected override void AddParsedSubObject(object obj);

3 [C++] protected: void AddParsedSubObject(Object\* obj);

4 [VB] Overrides Protected Sub AddParsedSubObject(ByVal obj As Object)

5 [JScript] protected override function AddParsedSubObject(obj : Object);

6 LoadViewState

7  
8 [C#] protected override void LoadViewState(object savedState);

9 [C++] protected: void LoadViewState(Object\* savedState);

10 [VB] Overrides Protected Sub LoadViewState(ByVal savedState As Object)

11 [JScript] protected override function LoadViewState(savedState : Object);

12  
13 *Description*

14 Load previously saved state. Overridden to synchronize Text property with  
15 LiteralContent.

16 RenderContents

17  
18 [C#] protected override void RenderContents(HtmlTextWriter writer);

19 [C++] protected: void RenderContents(HtmlTextWriter\* writer);

20 [VB] Overrides Protected Sub RenderContents(ByVal writer As HtmlTextWriter)

21 [JScript] protected override function RenderContents(writer : HtmlTextWriter);

22  
23 *Description*

24 Displays the **System.Web.UI.WebControls.HyperLink** on a page. The  
25 output stream to render on the client.

HyperLinkColumn class (System.Web.UI.WebControls)

TrackViewState

### *Description*

A column type for the **System.Web.UI.WebControls.DataGrid** control that contains a hyperlink for each item in the column.

Use the **System.Web.UI.WebControls.HyperLinkColumn** column type in a **System.Web.UI.WebControls.DataGrid** control to create a hyperlink for each row in the **System.Web.UI.WebControls.DataGrid** control. Set the **System.Web.UI.WebControls.HyperLinkColumn.Text** property to specify the caption text for the hyperlink. To specify the URL to navigate to when the hyperlink is clicked, set the **System.Web.UI.WebControls.HyperLinkColumn.NavigateUrl** property.

HyperLinkColumn

*Example Syntax:*

TrackViewState

```
[C#] public HyperLinkColumn();
```

```
[C++] public: HyperLinkColumn();
```

```
[VB] Public Sub New()
```

```
[JScript] public function HyperLinkColumn();
```

### *Description*

1        Initializes a new instance of the  
2        **System.Web.UI.WebControls.HyperLinkColumn** class.

3        Use this constructor to create and initialize a new instance of the  
4        **System.Web.UI.WebControls.HyperLinkColumn** class.

5        DataNavigateUrlField

6        TrackViewState

7  
8        [C#] public virtual string DataNavigateUrlField {get; set;}

9        [C++] public: \_\_property virtual String\* get\_DataNavigateUrlField();public:

10        \_\_property virtual void set\_DataNavigateUrlField(String\*);

11        [VB] Overridable Public Property DataNavigateUrlField As String

12        [JScript] public function get DataNavigateUrlField() : String;public function set

13        DataNavigateUrlField(String);

14  
15        *Description*

16        Gets or sets the field name from a data source to bind to the URL of the  
17        hyperlinks in **System.Web.UI.WebControls.HyperLinkColumn** column.

18        DataNavigateUrlFormatString

19        TrackViewState

20  
21        [C#] public virtual string DataNavigateUrlFormatString {get; set;}

22        [C++] public: \_\_property virtual String\*

23        get\_DataNavigateUrlFormatString();public: \_\_property virtual void

24        set\_DataNavigateUrlFormatString(String\*);

25        [VB] Overridable Public Property DataNavigateUrlFormatString As String

[JScript] public function get DataNavigateUrlFormatString() : String;public  
function set DataNavigateUrlFormatString(String);

*Description*

Gets or sets the string that specifies the display format for the URL of the  
hyperlinks in the **System.Web.UI.WebControls.HyperLinkColumn** column.

Use the  
**System.Web.UI.WebControls.HyperLinkColumn.DataNavigateUrlFormatStr  
ing** property to provide a custom format for the URL of the hyperlinks in the  
**System.Web.UI.WebControls.HyperLinkColumn** column.

DataTextField

TrackViewState

[C#] public virtual string DataTextField {get; set;}

[C++] public: \_\_property virtual String\* get\_DataTextField();public: \_\_property  
virtual void set\_DataTextField(String\*);

[VB] Overridable Public Property DataTextField As String

[JScript] public function get DataTextField() : String;public function set  
DataTextField(String);

*Description*

Gets or sets the field name from a data source to bind to the text caption of  
the hyperlinks in **System.Web.UI.WebControls.HyperLinkColumn** column.

DataTextFormatString

TrackViewState

```

1
2 [C#] public virtual string DataTextFormatString {get; set;}
3 [C++] public: __property virtual String* get_DataTextFormatString();public:
4 __property virtual void set_DataTextFormatString(String*);
5 [VB] Overridable Public Property DataTextFormatString As String
6 [JScript] public function get DataTextFormatString() : String;public function set
7 DataTextFormatString(String);
8

```

### *Description*

Gets or sets the string that specifies the display format for the text caption of the hyperlinks in the **System.Web.UI.WebControls.HyperLinkColumn** column.

Use the **System.Web.UI.WebControls.HyperLinkColumn.DataTextFormatString** property to provide a custom format for the items in the column.

DesignMode

FooterStyle

FooterText

HeaderImageUrl

HeaderStyle

HeaderText

IsTrackingViewState

ItemStyle

NavigateUrl

TrackViewState



1  
2  
3 *Description*

4 Gets or sets the URL to link to when a hyperlink in the column is clicked.

5 Use the **System.Web.UI.WebControls.HyperLinkColumn.NavigateUrl**  
6 property to specify the URL to navigate to when a hyperlink in the column is  
7 clicked.

8 Owner

9 SortExpression

10 Target

11 TrackViewState

12  
13  
14 *Description*

15 Gets or sets the target window or frame to display the Web page content  
16 linked to when the hyperlink in the column is clicked.

17 Use the **System.Web.UI.WebControls.HyperLinkColumn.Target**  
18 property to specify the frame or window that displays the Web page linked to  
19 when a hyperlink in the column is clicked.

20 Text

21 TrackViewState

22  
23 [C#] public virtual string Text {get; set;}

24 [C++] public: \_\_property virtual String\* get\_Text();public: \_\_property virtual

25 void set\_Text(String\*);

1 [VB] Overridable Public Property Text As String

2 [JScript] public function get Text() : String;public function set Text(String);

4 *Description*

5 Gets or sets the text caption to display for the hyperlinks in the column.

6 Use the **System.Web.UI.WebControls.HyperLinkColumn.Text** property  
7 to specify the text caption to display for the hyperlinks in the column.

8 ViewState

9 Visible

10 FormatDataNavigateUrlValue

12 [C#] protected virtual string FormatDataNavigateUrlValue(object dataUrlValue);

13 [C++] protected: virtual String\* FormatDataNavigateUrlValue(Object\*  
14 dataUrlValue);

15 [VB] Overridable Protected Function FormatDataNavigateUrlValue(ByVal  
16 dataUrlValue As Object) As String

17 [JScript] protected function FormatDataNavigateUrlValue(dataUrlValue : Object)  
18 : String;

20 *Description*

21 FormatDataTextValue

23 [C#] protected virtual string FormatDataTextValue(object dataTextValue);

24 [C++] protected: virtual String\* FormatDataTextValue(Object\* dataTextValue);

25 [VB] Overridable Protected Function FormatDataTextValue(ByVal

1 dataTextView As Object) As String  
2 [JScript] protected function FormatDataTextView(dataTextView : Object) :  
3 String;

4  
5 *Description*

6 Initialize

7  
8 [C#] public override void Initialize();  
9 [C++] public: void Initialize();  
10 [VB] Overrides Public Sub Initialize()  
11 [JScript] public override function Initialize();

12  
13 *Description*

14 InitializeCell

15  
16 [C#] public override void InitializeCell(TableCell cell, int columnIndex,  
17 ListItemType itemType);  
18 [C++] public: void InitializeCell(TableCell\* cell, int columnIndex, ListItemType  
19 itemType);  
20 [VB] Overrides Public Sub InitializeCell(ByVal cell As TableCell, ByVal  
21 columnIndex As Integer, ByVal itemType As ListItemType)  
22 [JScript] public override function InitializeCell(cell : TableCell, columnIndex : int,  
23 itemType : ListItemType);

24  
25 *Description*

1        Initializes the cell representing this column with the contained hyperlink.

2        HyperLinks are created for all items in the DataGrid except the Header and  
3        Footer items. The cell to be initialized. The index of the column that contains the  
4        cell. The type of item that the cell is part of.

5        HyperLinkControlBuilder class (System.Web.UI.WebControls)

6        TrackViewState

7  
8  
9        *Description*

10        Interacts with the parser to build a  
11        **System.Web.UI.WebControls.HyperLink** control.

12        To create a custom control builder for a  
13        **System.Web.UI.WebControls.HyperLink** derived control, you need to inherit  
14        from this class.

15        HyperLinkControlBuilder

16        *Example Syntax:*

17        TrackViewState

18  
19        [C#] public HyperLinkControlBuilder();

20        [C++] public: HyperLinkControlBuilder();

21        [VB] Public Sub New()

22        [JScript] public function HyperLinkControlBuilder();

23        ControlType

24        FChildrenAsProperties

25        FIsNonParserAccessor

1 HasAspCode  
2 ID  
3 InDesigner  
4 NamingContainerType  
5 Parser  
6 TagName  
7 AllowWhitespaceLiterals  
8

9 [C#] public override bool AllowWhitespaceLiterals();  
10 [C++] public: bool AllowWhitespaceLiterals();  
11 [VB] Overrides Public Function AllowWhitespaceLiterals() As Boolean  
12 [JScript] public override function AllowWhitespaceLiterals() : Boolean;  
13

14 *Description*

15 Gets a value that indicates whether white spaces are allowed in literals for  
16 this control.

17 *Return Value:* Overloaded to always returns **false** to indicate that white spaces are  
18 not allowed.

19 This method overrides the  
20 **System.Web.UI.ControlBuilder.AllowWhitespaceLiterals** property to ignore  
21 white space in the **System.Web.UI.WebControls.HyperLink** control.

22 Image class (System.Web.UI.WebControls)

23 ToString  
24  
25

1  
2  
3 *Description*

4 Displays an image on a Web page.

5 Use the **System.Web.UI.WebControls.Image** control to display an image  
6 on the Web page. The path to the displayed image is specified by setting the  
7 **System.Web.UI.WebControls.Image.ImageUrl** property. You can specify the  
8 text to display in place of image when the image is not available by setting the  
9 **System.Web.UI.WebControls.Image.AlternateText** property. The alignment of  
10 the image in relation to other elements on the Web page is specified by setting  
11 **System.Web.UI.WebControls.Image.ImageAlign** property.

12 Image

13 *Example Syntax:*

14 ToString

15  
16 [C#] public Image();  
17 [C++] public: Image();  
18 [VB] Public Sub New()  
19 [JScript] public function Image();  
20

21 *Description*

22 Initializes a new instance of the **System.Web.UI.WebControls.Image**  
23 class.

24 Use this constructor to create and initialize a new instance of the  
25 **System.Web.UI.WebControls.Image** class.

1       AccessKey  
2       AlternateText  
3       ToString

4  
5  
6       *Description*

7           Gets or sets the alternate text displayed in the  
8       **System.Web.UI.WebControls.Image** control when the image is unavailable.  
9       Browsers that support the ToolTips feature display this text as a ToolTip.  
10           Use this property to specify the text to display if the image specified in the  
11       **System.Web.UI.WebControls.Image.ImageUrl** property is not available. In  
12       browsers that support the ToolTips feature, this text also displays as a ToolTip.

13       Attributes  
14       BackColor  
15       BorderColor  
16       BorderStyle  
17       BorderWidth  
18       ChildControlsCreated  
19       ClientID  
20       Context  
21       Controls  
22       ControlStyle  
23       ControlStyleCreated  
24       CssClass  
25       Enabled

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

ToString

*Description*

Gets or sets a value indicating whether the control is enabled.

This property is inherited from the

**System.Web.UI.WebControls.WebControl** class and is not applicable to the  
**System.Web.UI.WebControls.Image** control.

EnableViewState

Events

Font

ToString

*Description*

Gets the font properties for the text associated with the control.

This property is inherited from the

**System.Web.UI.WebControls.WebControl** class and is not applicable to the  
**System.Web.UI.WebControls.Image** control.

ForeColor

HasChildViewState

Height

ID

ImageAlign

ToString



1  
2  
3 *Description*

4 Gets or sets the alignment of the **System.Web.UI.WebControls.Image**  
5 control in relation to other elements on the Web page.

6 Use the **System.Web.UI.WebControls.Image.ImageAlign** property to  
7 specify or determine the alignment of the image in relation to other elements on  
8 the Web page. The following table lists the possible alignments.

9 **ImageUrl**

10 **ToString**

11  
12 [C#] public virtual string ImageUrl {get; set;}

13 [C++] public: \_\_property virtual String\* get\_ImageUrl();public: \_\_property virtual  
14 void set\_ImageUrl(String\*);

15 [VB] Overridable Public Property ImageUrl As String

16 [JScript] public function get ImageUrl() : String;public function set  
17 ImageUrl(String);

18  
19 *Description*

20 Gets or sets the location of an image to display in the  
21 **System.Web.UI.WebControls.Image** control.

22 Use the **System.Web.UI.WebControls.Image.ImageUrl** property to  
23 specify the URL of an image to display in the  
24 **System.Web.UI.WebControls.Image** control. You can use a relative or an  
25 absolute URL. A relative URL relates the location of the image to the location of

1 the Web page without specifying a complete path on the server. The path is  
2 relative to the location of the Web page. This makes it easier to move the entire  
3 site to another directory on the server without updating the code. An absolute URL  
4 provides the complete path, so moving the site to another directory requires that  
5 you update the code.

6       IsTrackingViewState

7       NamingContainer

8       Page

9       Parent

10      Site

11      Style

12      TabIndex

13      TagKey

14      TagName

15      TemplateSourceDirectory

16      ToolTip

17      UniqueID

18      ViewState

19      ViewStateIgnoresCase

20      Visible

21      Width

22      AddAttributesToRender

23  
24 [C#] protected override void AddAttributesToRender(HtmlTextWriter writer);

25 [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);

1 [VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As  
2 HtmlTextWriter)

3 [JScript] protected override function AddAttributesToRender(writer :  
4 HtmlTextWriter);

5  
6 *Description*

7 Adds the attributes of an **System.Web.UI.WebControls.Image** to the  
8 output stream for rendering on the client. A **System.Web.UI.HtmlTextWriter**  
9 that contains the output stream to render on the client browser.

10 **RenderContents**

11  
12 [C#] protected override void RenderContents(HtmlTextWriter writer);

13 [C++] protected: void RenderContents(HtmlTextWriter\* writer);

14 [VB] Overrides Protected Sub RenderContents(ByVal writer As HtmlTextWriter)

15 [JScript] protected override function RenderContents(writer : HtmlTextWriter);

16  
17 *Description*

18 **ImageAlign** enumeration (System.Web.UI.WebControls)

19 **TrackViewState**

20  
21  
22 *Description*

23 Specifies the alignment of an image in relation to the text of a Web page.

24 The **System.Web.UI.WebControls.ImageAlign** enumeration represents  
25 the alignment options for an image relative to the text of a Web page.

TrackViewState

[C#] public const ImageAlign AbsBottom;  
[C++] public: const ImageAlign AbsBottom;  
[VB] Public Const AbsBottom As ImageAlign  
[JScript] public var AbsBottom : ImageAlign;

*Description*

The lower edge of the image is aligned with the lower edge of the largest element on the same line.

TrackViewState

[C#] public const ImageAlign AbsMiddle;  
[C++] public: const ImageAlign AbsMiddle;  
[VB] Public Const AbsMiddle As ImageAlign  
[JScript] public var AbsMiddle : ImageAlign;

*Description*

The middle of the image is aligned with the middle of the largest element on the same line.

TrackViewState

[C#] public const ImageAlign Baseline;  
[C++] public: const ImageAlign Baseline;  
[VB] Public Const Baseline As ImageAlign

1 [JScript] public var Baseline : ImageAlign;

2  
3 *Description*

4 The lower edge of the image is aligned with the lower edge of the first line  
5 of text.

6 TrackViewState

7  
8 [C#] public const ImageAlign Bottom;

9 [C++] public: const ImageAlign Bottom;

10 [VB] Public Const Bottom As ImageAlign

11 [JScript] public var Bottom : ImageAlign;

12  
13 *Description*

14 The lower edge of the image is aligned with the lower edge of the first line  
15 of text.

16 TrackViewState

17  
18 [C#] public const ImageAlign Left;

19 [C++] public: const ImageAlign Left;

20 [VB] Public Const Left As ImageAlign

21 [JScript] public var Left : ImageAlign;

22  
23 *Description*

24 The image is aligned on the left edge of the Web page with text wrapping  
25 on the right.

## TrackViewState

```
[C#] public const ImageAlign Middle;  
[C++] public: const ImageAlign Middle;  
[VB] Public Const Middle As ImageAlign  
[JScript] public var Middle : ImageAlign;
```

### *Description*

The middle of the image is aligned with the lower edge of the first line of text.

## TrackViewState

```
[C#] public const ImageAlign NotSet;  
[C++] public: const ImageAlign NotSet;  
[VB] Public Const NotSet As ImageAlign  
[JScript] public var NotSet : ImageAlign;
```

### *Description*

The alignment is not set.

## TrackViewState

```
[C#] public const ImageAlign Right;  
[C++] public: const ImageAlign Right;  
[VB] Public Const Right As ImageAlign  
[JScript] public var Right : ImageAlign;
```

1  
2 *Description*

3 The image is aligned on the right edge of the Web page with text wrapping  
4 on the left.

5 TrackViewState

6  
7 [C#] public const ImageAlign TextTop;

8 [C++] public: const ImageAlign TextTop;

9 [VB] Public Const TextTop As ImageAlign

10 [JScript] public var TextTop : ImageAlign;

11  
12 *Description*

13 The upper edge of the image is aligned with the upper edge of the highest  
14 text on the same line.

15 TrackViewState

16  
17 [C#] public const ImageAlign Top;

18 [C++] public: const ImageAlign Top;

19 [VB] Public Const Top As ImageAlign

20 [JScript] public var Top : ImageAlign;

21  
22 *Description*

23 The upper edge of the image is aligned with the upper edge of the highest  
24 element on the same line.

25 ImageButton class (System.Web.UI.WebControls)

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

ToString

*Description*

A control that displays an image and responds to mouse clicks on the image.

Use the **System.Web.UI.WebControls.ImageButton** control to display an image that responds to mouse clicks.

ImageButton

*Example Syntax:*

ToString

```
[C#] public ImageButton();  
[C++] public: ImageButton();  
[VB] Public Sub New()  
[JScript] public function ImageButton();
```

*Description*

Initializes a new instance of the **System.Web.UI.WebControls.ImageButton** class.

Use this constructor to create and initialize a new instance of the **System.Web.UI.WebControls.ImageButton** class.

AccessKey

AlternateText

Attributes



1 BackColor  
2 BorderColor  
3 BorderStyle  
4 BorderWidth  
5 CausesValidation  
6 ToString

7  
8  
9 *Description*

10 Gets or sets a value indicating whether validation is performed when the  
11 **System.Web.UI.WebControls.ImageButton** control is clicked.

12 By default, page validation is performed when a  
13 **System.Web.UI.WebControls.ImageButton** control is clicked. Page validation  
14 determines whether the input controls associated with a validation control on the  
15 page all pass the validation rules specified by the validation control.

16 ChildControlsCreated  
17 ClientID  
18 CommandArgument  
19 ToString

20  
21  
22 *Description*

23 Gets or sets an optional argument that provides additional information  
24 about the **System.Web.UI.WebControls.ImageButton.CommandName**  
25 property.

Sometimes, multiple **System.Web.UI.WebControls.ImageButton** controls are related and share the same value for the **System.Web.UI.WebControls.ImageButton.CommandName** property, such as **Sort**. Use this property to supplement the **System.Web.UI.WebControls.ImageButton.CommandName** property with additional information about the command to perform, such as **Ascending**. The values of the **System.Web.UI.WebControls.ImageButton.CommandName** and **System.Web.UI.WebControls.ImageButton.CommandArgument** properties are typically used in the **System.Web.UI.WebControls.ImageButton.OnCommand(System.Web.UI.WebControls.CommandEventArgs)** event handler to determine the action to perform when the **System.Web.UI.WebControls.ImageButton** control is clicked.

**CommandName**

**ToString**

[C#] public string CommandName {get; set;}

[C++] public: \_\_property String\* get\_CommandName();public: \_\_property void set\_CommandName(String\*);

[VB] Public Property CommandName As String

[JScript] public function get CommandName() : String;public function set CommandName(String);

*Description*

1 Gets or sets the command name associated with the  
2 **System.Web.UI.WebControls.ImageButton** control.

3 Use this property to specify the command to perform when the  
4 **System.Web.UI.WebControls.ImageButton** control is clicked, such as **Sort** ,  
5 **Cancel** , and **Edit** . This allows multiple  
6 **System.Web.UI.WebControls.ImageButton** controls to be placed on the same  
7 Web page. The value in this property can then be programmatically identified in  
8 the  
9 **System.Web.UI.WebControls.ImageButton.OnCommand(System.Web.UI.W**  
10 **ebControls.CommandEventArgs)** event handler to determine the appropriate  
11 action to perform when each **System.Web.UI.WebControls.ImageButton**  
12 control is clicked.

13 Context

14 Controls

15 ControlStyle

16 ControlStyleCreated

17 CssClass

18 Enabled

19 EnableViewState

20 Events

21 Font

22 ForeColor

23 HasChildViewState

24 Height

25 ID

1	ImageAlign
2	ImageUrl
3	IsTrackingViewState
4	NamingContainer
5	Page
6	Parent
7	Site
8	Style
9	TabIndex
10	TagKey
11	ToString

14 *Description*

15 Gets a value that represents the tag HtmlTextWriterTag.Input. This  
 16 property is read-only.

17 Overrides the base implementation and returns the tag  
 18 HtmlTextWriterTag.Input instead of the default HtmlTextWriterTag.Image Gets a  
 19 value that represents the tag HtmlTextWriterTag.Input. This property is read-only.

20	TagName
21	TemplateSourceDirectory
22	ToolTip
23	UniqueID
24	ViewState
25	ViewStateIgnoresCase

1 Visible  
2 Width  
3 ToString  
4  
5

6 *Description*

7 Occurs when the **System.Web.UI.WebControls.ImageButton** is clicked.  
8 The **System.Web.UI.WebControls.ImageButton.Click** event is raised  
9 when the **System.Web.UI.WebControls.ImageButton** control is clicked.

10 ToString  
11

12 [C#] public event CommandEventHandler Command;  
13 [C++] public: \_\_event CommandEventHandler\* Command;  
14 [VB] Public Event Command As CommandEventHandler  
15

16 *Description*

17 Occurs when the **System.Web.UI.WebControls.ImageButton** is clicked.  
18 The **System.Web.UI.WebControls.ImageButton.Command** event is  
19 raised when an **System.Web.UI.WebControls.ImageButton** control is clicked.

20 AddAttributesToRender  
21

22 [C#] protected override void AddAttributesToRender(HtmlTextWriter writer);  
23 [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);  
24 [VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As  
25 HtmlTextWriter)

1 [JScript] protected override function AddAttributesToRender(writer :  
2 HtmlTextWriter);

3  
4 *Description*

5 Adds the attributes of an **System.Web.UI.WebControls.ImageButton** to  
6 the output stream for rendering on the client. The output stream to render on the  
7 client.

8 **OnClick**

9  
10 [C#] protected virtual void OnClick(ImageClickEventArgs e);

11 [C++] protected: virtual void OnClick(ImageClickEventArgs\* e);

12 [VB] Overridable Protected Sub OnClick(ByVal e As ImageClickEventArgs)

13 [JScript] protected function OnClick(e : ImageClickEventArgs);

14  
15 *Description*

16 Raises the **System.Web.UI.WebControls.ImageButton.Click** event and  
17 allows you to handle the **System.Web.UI.WebControls.ImageButton.Click**  
18 event directly.

19 The **System.Web.UI.WebControls.ImageButton.Click** event is raised  
20 when the **System.Web.UI.WebControls.ImageButton** control is clicked. By  
21 using the  
22 **System.Web.UI.WebControls.ImageButton.OnClick(System.Web.UI.ImageC**  
23 **lickEventArgs)** event handler, you can programmatically determine the  
24 coordinates where the image is clicked. You can then code a response, based on  
25 the values of these coordinates. Note the origin (0, 0) is located at the upper left

corner of the image. A **System.Web.UI.ImageClickEventArgs** that contains the event data.

### OnCommand

[C#] protected virtual void OnCommand(CommandEventArgs e);

[C++] protected: virtual void OnCommand(CommandEventArgs\* e);

[VB] Overridable Protected Sub OnCommand(ByVal e As CommandEventArgs)

[JScript] protected function OnCommand(e : CommandEventArgs);

### Description

Raises the **System.Web.UI.WebControls.ImageButton.Command** event and allows you to handle the **System.Web.UI.WebControls.ImageButton.Command** event directly.

The **System.Web.UI.WebControls.ImageButton.Command** event is raised when the **System.Web.UI.WebControls.ImageButton** control is clicked.

The

**System.Web.UI.WebControls.ImageButton.OnCommand(System.Web.UI.WebControls.CommandEventArgs)** event handler is used to make the

**System.Web.UI.WebControls.ImageButton** control behave like a command button. A command name can be associated with the control by using the

**System.Web.UI.WebControls.ImageButton.CommandName** property. This allows multiple **System.Web.UI.WebControls.ImageButton** controls to be placed on the Web page. The value in this property can then be programmatically identified in the

**System.Web.UI.WebControls.ImageButton.OnCommand(System.Web.UI.W**

**ebControls.CommandEventArgs**) event handler to determine the appropriate action to perform when each **System.Web.UI.WebControls.ImageButton** control is clicked. The **System.Web.UI.WebControls.ImageButton.CommandArgument** property can also be used to pass additional information about the command, such as specifying ascending order. A **System.Web.UI.WebControls.CommandEventArgs** that contains the event data.

#### OnPreRender

[C#] protected override void OnPreRender(EventArgs e);  
 [C++] protected: void OnPreRender(EventArgs\* e);  
 [VB] Overrides Protected Sub OnPreRender(ByVal e As EventArgs)  
 [JScript] protected override function OnPreRender(e : EventArgs);

#### *Description*

Determine if the image has been clicked prior to rendering on the client.

#### IPostBackDataHandler.LoadPostData

[C#] bool IPostBackDataHandler.LoadPostData(string postDataKey, NameValueCollection postCollection);  
 [C++] bool IPostBackDataHandler::LoadPostData(String\* postDataKey, NameValueCollection\* postCollection);  
 [VB] Function LoadPostData(ByVal postDataKey As String, ByVal postCollection As NameValueCollection) As Boolean Implements IPostBackDataHandler.LoadPostData



1 [JScript] function IPostBackDataHandler.LoadPostData(postDataKey : String,  
2 postCollection : NameValueCollection) : Boolean;

3 IPostBackDataHandler.RaisePostDataChangedEvent

5 [C#] void IPostBackDataHandler.RaisePostDataChangedEvent();

6 [C++] void IPostBackDataHandler::RaisePostDataChangedEvent();

7 [VB] Sub RaisePostDataChangedEvent() Implements

8 IPostBackDataHandler.RaisePostDataChangedEvent

9 [JScript] function IPostBackDataHandler.RaisePostDataChangedEvent();

10 IPostBackEventHandler.RaisePostBackEvent

12 [C#] void IPostBackEventHandler.RaisePostBackEvent(string eventArgument);

13 [C++] void IPostBackEventHandler::RaisePostBackEvent(String\*  
14 eventArgument);

15 [VB] Sub RaisePostBackEvent(ByVal eventArgument As String) Implements

16 IPostBackEventHandler.RaisePostBackEvent

17 [JScript] function IPostBackEventHandler.RaisePostBackEvent(eventArgument :  
18 String);

19 IRepeatInfoUser interface (System.Web.UI.WebControls)

20 TrackViewState

23 *Description*

24 Specifies a contract for implementing

25 **System.Web.UI.WebControls.Repeater** objects in list controls.

1       HasFooter  
2       TrackViewState  
3  
4   [C#] bool HasFooter {get;}  
5   [C++] bool get\_HasFooter();  
6   [VB] ReadOnly Property HasFooter As Boolean  
7   [JScript] abstract function get HasFooter() : Boolean;

8  
9   *Description*

10       Indicates whether the Repeater contains a footer item.

11       HasHeader

12       TrackViewState  
13

14   [C#] bool HasHeader {get;}  
15   [C++] bool get\_HasHeader();  
16   [VB] ReadOnly Property HasHeader As Boolean  
17   [JScript] abstract function get HasHeader() : Boolean;

18  
19   *Description*

20       Indicates whether the **System.Web.UI.WebControls.Repeater** contains a  
21   header item.

22       HasSeparators

23       TrackViewState  
24

25   [C#] bool HasSeparators {get;}

1 [C++] bool get\_HasSeparators();

2 [VB] ReadOnly Property HasSeparators As Boolean

3 [JScript] abstract function get HasSeparators() : Boolean;

4  
5 *Description*

6 Indicates whether the Repeater contains separator items.

7 RepeatedItemCount

8 TrackViewState

9  
10 [C#] int RepeatedItemCount {get;}

11 [C++] int get\_RepeatedItemCount();

12 [VB] ReadOnly Property RepeatedItemCount As Integer

13 [JScript] abstract function get RepeatedItemCount() : int;

14  
15 *Description*

16 Specifies the item count of the Repeater.

17 GetItemStyle

18  
19 [C#] Style GetItemStyle(ListItemType itemType, int repeatIndex);

20 [C++] Style\* GetItemStyle(ListItemType itemType, int repeatIndex);

21 [VB] Function GetItemStyle(ByVal itemType As ListItemType, ByVal  
22 repeatIndex As Integer) As Style

23 [JScript] function GetItemStyle(itemType : ListItemType, repeatIndex : int) :  
24 Style;

1  
2 *Description*

3       Retrieves the item style with the specified item type and location within the  
4 **System.Web.UI.WebControls.Repeater** .

5 *Return Value:* A **System.Web.UI.WebControls.Style** that represents the Repeater  
6 item style. A **System.Web.UI.WebControls.ListItemType** that represents the  
7 specified type of the **System.Web.UI.WebControls.Repeater** item. An ordinal  
8 index that specifies the location of the item within the  
9 **System.Web.UI.WebControls.Repeater**.

10       RenderItem

11  
12 [C#] void RenderItem(ListItemType itemType, int repeatIndex, RepeatInfo  
13 repeatInfo, HtmlTextWriter writer);

14 [C++] void RenderItem(ListItemType itemType, int repeatIndex, RepeatInfo\*  
15 repeatInfo, HtmlTextWriter\* writer);

16 [VB] Sub RenderItem(ByVal itemType As ListItemType, ByVal repeatIndex As  
17 Integer, ByVal repeatInfo As RepeatInfo, ByVal writer As HtmlTextWriter)

18 [JScript] function RenderItem(itemType : ListItemType, repeatIndex : int,  
19 repeatInfo : RepeatInfo, writer : HtmlTextWriter);

20  
21 *Description*

22       Renders the **System.Web.UI.WebControls.Repeater** item with the  
23 specified information. A **System.Web.UI.WebControls.ListItemType** that  
24 represents the specified type of the **System.Web.UI.WebControls.Repeater** item.  
25 An ordinal index that specifies the location of the item within the

1 **System.Web.UI.WebControls.Repeater**. A

2 **System.Web.UI.WebControls.RepeatInfo** that represents the information used  
3 to render items using a **System.Web.UI.WebControls.Repeater** . The output  
4 stream that renders HTML content to the client.

5 Label class (System.Web.UI.WebControls)

6 RenderItem

7  
8  
9 *Description*

10 Represents a label control, which displays text on a Web page.

11 Use the **System.Web.UI.WebControls.Label** control to display text in a  
12 set location on the page. Unlike static text, you can customize the displayed text  
13 through the **System.Web.UI.WebControls.Label.Text** property.

14 Label

15 *Example Syntax:*

16 RenderItem

17  
18 [C#] public Label();

19 [C++] public: Label();

20 [VB] Public Sub New()

21 [JScript] public function Label(); Initializes a new instance of the

22 **System.Web.UI.WebControls.Label** class.

23  
24 *Description*

1        Initializes a new instance of the **System.Web.UI.WebControls.Label**  
2 class.

3        Use this constructor to create and initialize a new instance of the  
4 **System.Web.UI.WebControls.Label** class.

5        AccessKey

6        Attributes

7        BackColor

8        BorderColor

9        BorderStyle

10       BorderWidth

11       ChildControlsCreated

12       ClientID

13       Context

14       Controls

15       ControlStyle

16       ControlStyleCreated

17       CssClass

18       Enabled

19       EnableViewState

20       Events

21       Font

22       ForeColor

23       HasChildViewState

24       Height

25       ID

1	IsTrackingViewState
2	NamingContainer
3	Page
4	Parent
5	Site
6	Style
7	TabIndex
8	TagKey
9	TagName
10	TemplateSourceDirectory
11	Text
12	RenderItem

13  
14  
15 *Description*

16 Gets or sets the text content of the **System.Web.UI.WebControls.Label**  
17 control.

18 Use the **System.Web.UI.WebControls.Label.Text** property to specify or  
19 determine the text content of the **System.Web.UI.WebControls.Label** control.  
20 This property is commonly used to programmatically customize the text that is  
21 displayed in the **System.Web.UI.WebControls.Label** control.

22	ToolTip
23	UniqueID
24	ViewState
25	ViewStateIgnoresCase

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

Visible

Width

AddParsedSubObject

```
[C#] protected override void AddParsedSubObject(object obj);  
[C++] protected: void AddParsedSubObject(Object* obj);  
[VB] Overrides Protected Sub AddParsedSubObject(ByVal obj As Object)  
[JScript] protected override function AddParsedSubObject(obj : Object);
```

*Description*

LoadViewState

```
[C#] protected override void LoadViewState(object savedState);  
[C++] protected: void LoadViewState(Object* savedState);  
[VB] Overrides Protected Sub LoadViewState(ByVal savedState As Object)  
[JScript] protected override function LoadViewState(savedState : Object);
```

*Description*

Load previously saved state. Overridden to synchronize Text property with LiteralContent. Represents the previously saved state.

RenderContents

```
[C#] protected override void RenderContents(HtmlTextWriter writer);  
[C++] protected: void RenderContents(HtmlTextWriter* writer);  
[VB] Overrides Protected Sub RenderContents(ByVal writer As HtmlTextWriter)
```



[JScript] protected override function RenderContents(writer : HtmlTextWriter);

*Description*

Renders the contents of the **System.Web.UI.WebControls.Label** into the specified writer. The output stream that renders HTML content to the client.

LabelControlBuilder class (System.Web.UI.WebControls)

TrackViewState

*Description*

Interacts with the parser to build a **System.Web.UI.WebControls.Label** control.

To create a custom control builder for a **System.Web.UI.WebControls.Label** derived control, you need to inherit from this class.

LabelControlBuilder

*Example Syntax:*

TrackViewState

[C#] public LabelControlBuilder();

[C++] public: LabelControlBuilder();

[VB] Public Sub New()

[JScript] public function LabelControlBuilder();

ControlType

FChildrenAsProperties

1        **IsNonParserAccessor**  
2        **HasAspCode**  
3        **ID**  
4        **InDesigner**  
5        **NamingContainerType**  
6        **Parser**  
7        **TagName**  
8        **AllowWhitespaceLiterals**

9  
10    **[C#] public override bool AllowWhitespaceLiterals();**  
11    **[C++] public: bool AllowWhitespaceLiterals();**  
12    **[VB] Overrides Public Function AllowWhitespaceLiterals() As Boolean**  
13    **[JScript] public override function AllowWhitespaceLiterals() : Boolean;**

14  
15    *Description*

16        Specifies whether white space literals are allowed.

17    *Return Value:* **false** for all cases.

18        This method overrides the

19    **System.Web.UI.ControlBuilder.AllowWhitespaceLiterals** property to ignore  
20    white space in the **System.Web.UI.WebControls.Label** control.

21        LinkButton class (System.Web.UI.WebControls)

22        ToString

23  
24  
25    *Description*

1 Displays a hyperlink style button control on a Web page.  
2 Use the **System.Web.UI.WebControls.LinkButton** control to create a  
3 hyperlink style button on the Web page. You can create either a submit button or a  
4 command button.

5 **LinkButton**

6 *Example Syntax:*

7 **ToString**

8  
9 [C#] public LinkButton();

10 [C++] public: LinkButton();

11 [VB] Public Sub New()

12 [JScript] public function LinkButton();

13  
14 *Description*

15 Initializes a new instance of the  
16 **System.Web.UI.WebControls.LinkButton** class.

17 Use this constructor to create and initialize a new instance of the  
18 **System.Web.UI.WebControls.LinkButton** control.

19 **AccessKey**

20 **Attributes**

21 **BackColor**

22 **BorderColor**

23 **BorderStyle**

24 **BorderWidth**

25 **CausesValidation**

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

ToString

*Description*

Gets or sets a value indicating whether validation is performed when the **System.Web.UI.WebControls.LinkButton** control is clicked.

By default, page validation is performed when a **System.Web.UI.WebControls.LinkButton** control is clicked. Page validation determines whether the input controls associated with a validation control on the page all pass the validation rules specified by the validation control.

ChildControlsCreated

ClientID

CommandArgument

ToString

*Description*

Gets or sets an optional argument passed to the **System.Web.UI.WebControls.LinkButton.Command** event handler along with the associated **System.Web.UI.WebControls.LinkButton.CommandName** property.

Use the **System.Web.UI.WebControls.LinkButton.CommandArgument** property to specify an argument that complements the **System.Web.UI.WebControls.LinkButton.CommandName** property.

CommandName

100720 696666

ToString

[C#] public string CommandName {get; set;}

[C++] public: \_\_property String\* get\_CommandName();public: \_\_property void  
set\_CommandName(String\*);

[VB] Public Property CommandName As String

[JScript] public function get CommandName() : String;public function set  
CommandName(String);

### *Description*

Gets or sets the command name associated with the **System.Web.UI.WebControls.LinkButton** control. This value is passed to the **System.Web.UI.WebControls.LinkButton.Command** event handler along with the **System.Web.UI.WebControls.LinkButton.CommandArgument** property.

When you have multiple **System.Web.UI.WebControls.LinkButton** controls on a Web page, use the **System.Web.UI.WebControls.LinkButton.CommandName** property to specify or determine the command name associated with the **System.Web.UI.WebControls.LinkButton** control clicked. You can set the **System.Web.UI.WebControls.LinkButton.CommandName** property with any string that identifies the command to perform. You can then programmatically determine the command name of the **System.Web.UI.WebControls.LinkButton** control and perform the appropriate actions.

Context

Controls



1  
2  
3 *Description*

4 Gets or sets the text caption displayed on the  
5 **System.Web.UI.WebControls.LinkButton** control.

6 Use the **System.Web.UI.WebControls.LinkButton.Text** property to  
7 specify or determine the caption to display on the  
8 **System.Web.UI.WebControls.LinkButton** control.

9 ToolTip

10 UniqueID

11 ViewState

12 ViewStateIgnoresCase

13 Visible

14 Width

15 ToString

16  
17  
18 *Description*

19 Occurs when the **System.Web.UI.WebControls.LinkButton** control is  
20 clicked.

21 The **System.Web.UI.WebControls.LinkButton.Click** event is raised  
22 when the **System.Web.UI.WebControls.LinkButton** control is clicked. This  
23 event is commonly used when no command name is associated with the  
24 **System.Web.UI.WebControls.LinkButton** control, such as a submit button.

25 ToString

1  
2 [C#] public event CommandEventHandler Command;  
3 [C++] public: \_\_event CommandEventHandler\* Command;  
4 [VB] Public Event Command As CommandEventHandler  
5

6 *Description*

7 Occurs when the **System.Web.UI.WebControls.Button** control is clicked.  
8 The **System.Web.UI.WebControls.LinkButton.Command** event is raised  
9 when the **System.Web.UI.WebControls.LinkButton** control is clicked. This  
10 event is commonly used when a command name, such as **Sort** , is associated with  
11 the **System.Web.UI.WebControls.LinkButton** control. This allows you to create  
12 multiple **System.Web.UI.WebControls.LinkButton** controls on a Web page and  
13 programmatically determine which **System.Web.UI.WebControls.LinkButton**  
14 control is clicked.

15 **AddAttributesToRender**  
16

17 [C#] protected override void AddAttributesToRender(HtmlTextWriter writer);  
18 [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);  
19 [VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As  
20 HtmlTextWriter)  
21 [JScript] protected override function AddAttributesToRender(writer :  
22 HtmlTextWriter);  
23

24 *Description*

25 Render the attributes on the begin tag.



## AddParsedSubObject

[C#] protected override void AddParsedSubObject(object obj);  
[C++] protected: void AddParsedSubObject(Object\* obj);  
[VB] Overrides Protected Sub AddParsedSubObject(ByVal obj As Object)  
[JScript] protected override function AddParsedSubObject(obj : Object);

### *Description*

## LoadViewState

[C#] protected override void LoadViewState(object savedState);  
[C++] protected: void LoadViewState(Object\* savedState);  
[VB] Overrides Protected Sub LoadViewState(ByVal savedState As Object)  
[JScript] protected override function LoadViewState(savedState : Object);

### *Description*

Load previously saved state. Overridden to synchronize Text property with LiteralContent.

## OnClick

[C#] protected virtual void OnClick(EventArgs e);  
[C++] protected: virtual void OnClick(EventArgs\* e);  
[VB] Overridable Protected Sub OnClick(ByVal e As EventArgs)  
[JScript] protected function OnClick(e : EventArgs);

1  
2 *Description*

3       Raises the **System.Web.UI.WebControls.LinkButton.Click** event of the  
4 **System.Web.UI.WebControls.LinkButton** control.

5       The **System.Web.UI.WebControls.LinkButton.Click** event is raised  
6 when the **System.Web.UI.WebControls.LinkButton** control is clicked. This  
7 event is commonly used when no command name is associated with the  
8 **System.Web.UI.WebControls.LinkButton** control, such as a submit button. A  
9 **System.EventArgs** that contains the event data.

10       **OnCommand**

11  
12 [C#] protected virtual void OnCommand(CommandEventArgs e);  
13 [C++] protected: virtual void OnCommand(CommandEventArgs\* e);  
14 [VB] Overridable Protected Sub OnCommand(ByVal e As CommandEventArgs)  
15 [JScript] protected function OnCommand(e : CommandEventArgs);  
16

17 *Description*

18       Raises the **System.Web.UI.WebControls.LinkButton.Command** event  
19 of the **System.Web.UI.WebControls.LinkButton** control.

20       The **System.Web.UI.WebControls.LinkButton.Command** event is raised  
21 when the **System.Web.UI.WebControls.LinkButton** control is clicked. This  
22 event is commonly used when a command name, such as **Sort** , is associated with  
23 the **System.Web.UI.WebControls.LinkButton** control. This allows you to create  
24 multiple **System.Web.UI.WebControls.LinkButton** controls on a Web page and  
25 programmatically determine which **System.Web.UI.WebControls.LinkButton**

control is clicked. A **System.Web.UI.WebControls.CommandEventArgs** that contains the event data.

**RenderContents**

[C#] protected override void RenderContents(HtmlTextWriter writer);

[C++] protected: void RenderContents(HtmlTextWriter\* writer);

[VB] Overrides Protected Sub RenderContents(ByVal writer As HtmlTextWriter)

[JScript] protected override function RenderContents(writer : HtmlTextWriter);

### *Description*

The output stream that renders HTML content to the client.

**IPostBackEventHandler.RaisePostBackEvent**

[C#] void IPostBackEventHandler.RaisePostBackEvent(string eventArgument);

[C++] void IPostBackEventHandler::RaisePostBackEvent(String\* eventArgument);

[VB] Sub RaisePostBackEvent(ByVal eventArgument As String) Implements IPostBackEventHandler.RaisePostBackEvent

[JScript] function IPostBackEventHandler.RaisePostBackEvent(eventArgument : String);

**LinkButtonControlBuilder** class (System.Web.UI.WebControls)

**TrackViewState**

### *Description*

1 Interacts with the parser to build a  
2 **System.Web.UI.WebControls.LinkButton** control.

3 To create a custom control builder for a  
4 **System.Web.UI.WebControls.LinkButton** derived control, you need to inherit  
5 from this class.

6 **LinkButtonControlBuilder**

7 *Example Syntax:*

8 **TrackViewState**

9  
10 [C#] public LinkButtonControlBuilder();

11 [C++] public: LinkButtonControlBuilder();

12 [VB] Public Sub New()

13 [JScript] public function LinkButtonControlBuilder();

14 **ControlType**

15 **FChildrenAsProperties**

16 **FIsNonParserAccessor**

17 **HasAspCode**

18 **ID**

19 **InDesigner**

20 **NamingContainerType**

21 **Parser**

22 **TagName**

23 **AllowWhitespaceLiterals**

24  
25 [C#] public override bool AllowWhitespaceLiterals();

1 [C++] public: bool AllowWhitespaceLiterals();

2 [VB] Overrides Public Function AllowWhitespaceLiterals() As Boolean

3 [JScript] public override function AllowWhitespaceLiterals() : Boolean;

4  
5 *Description*

6 Specifies whether white space literals are allowed.

7 *Return Value:* **false** for all cases.

8 This method overrides

9 **System.Web.UI.ControlBuilder.AllowWhitespaceLiterals** to ignore white  
10 space in the **System.Web.UI.WebControls.LinkButton** control.

11 **ListBox** class (**System.Web.UI.WebControls**)

12 **ToString**

13  
14  
15 *Description*

16 Represents a list box control that allows single or multiple item selection.

17 Use the **System.Web.UI.WebControls.ListBox** control to create a list  
18 control that allows single or multiple item selection. Use the  
19 **System.Web.UI.WebControls.ListBox.Rows** property to specify the height of  
20 the control. To enable multiple item selection, set the  
21 **System.Web.UI.WebControls.ListBox.SelectionMode** property to  
22 **ListSelectionMode.Multiple** .

23 **ListBox**

24 *Example Syntax:*

25 **ToString**

1  
2 [C#] public ListBox();  
3 [C++] public: ListBox();  
4 [VB] Public Sub New()  
5 [JScript] public function ListBox();  
6

7 *Description*

8       Initializes a new instance of the **System.Web.UI.WebControls.ListBox**  
9 class.

10       Use this constructor to create and initialize a new instance of the  
11 **System.Web.UI.WebControls.ListBox** class.

12       AccessKey

13       Attributes

14       AutoPostBack

15       BackColor

16       BorderColor

17       ToString

18  
19  
20 *Description*

21       Gets or sets the border color of the control.

22       The **System.Web.UI.WebControls.ListBox.BorderColor** property is  
23 inherited from the **System.Web.UI.WebControls.WebControl** class and is not  
24 applicable to the **System.Web.UI.WebControls.ListBox** control.

25       BorderStyle

ToString

[C#] public override BorderStyle BorderStyle {get; set;}

[C++] public: \_\_property virtual BorderStyle get\_BorderStyle();public:  
\_\_property virtual void set\_BorderStyle(BorderStyle);

[VB] Overrides Public Property BorderStyle As BorderStyle

[JScript] public function get BorderStyle() : BorderStyle;public function set  
BorderStyle(BorderStyle);

#### *Description*

Gets or sets the border style of the control.

The **System.Web.UI.WebControls.ListBox.BorderStyle** property is inherited from the **System.Web.UI.WebControls.WebControl** class and is not applicable to the **System.Web.UI.WebControls.ListBox** control.

BorderWidth

ToString

[C#] public override Unit BorderWidth {get; set;}

[C++] public: \_\_property virtual Unit get\_BorderWidth();public: \_\_property  
virtual void set\_BorderWidth(Unit);

[VB] Overrides Public Property BorderWidth As Unit

[JScript] public function get BorderWidth() : Unit;public function set  
BorderWidth(Unit);

#### *Description*

Gets or sets the border width for the control.

The **System.Web.UI.WebControls.ListBox.BorderWidth** property is inherited from the **System.Web.UI.WebControls.WebControl** class and is not applicable to the **System.Web.UI.WebControls.ListBox** control.

ChildControlsCreated

ClientID

Context

Controls

ControlStyle

ControlStyleCreated

CssClass

DataMember

DataSource

DataTextField

DataTextFormatString

DataValueField

Enabled

EnableViewState

Events

Font

ForeColor

HasChildViewState

Height

ID

IsTrackingViewState



1 Items  
2 NamingContainer  
3 Page  
4 Parent  
5 Rows  
6 ToString

7  
8  
9 *Description*

10 Gets or sets the number of rows displayed in the  
11 **System.Web.UI.WebControls.ListBox** control.

12 Use the **System.Web.UI.WebControls.ListBox.Rows** property to specify  
13 the number of rows to display in the **System.Web.UI.WebControls.ListBox**  
14 control.

15 SelectedIndex  
16 SelectedItem  
17 SelectionMode  
18 ToString

19  
20  
21 *Description*

22 Gets or sets the selection mode of the  
23 **System.Web.UI.WebControls.ListBox** control.

24 Use the **System.Web.UI.WebControls.ListBox.SelectionMode** property  
25 to specify the mode behavior of the **System.Web.UI.WebControls.ListBox**

control. Setting this property to **ListSelectionMode.Single** indicates only a single item can be selected from the **System.Web.UI.WebControls.ListBox** control, while **ListSelectionMode.Multiple** specifies multiple items can be selected.

Site

Style

TabIndex

TagKey

TagName

TemplateSourceDirectory

ToolTip

ToString

#### *Description*

Gets or sets the ToolTip text displayed when the mouse pointer rests over the control.

The **System.Web.UI.WebControls.ListBox.ToolTip** property is inherited from the **System.Web.UI.WebControls.WebControl** class and is not applicable to the **System.Web.UI.WebControls.ListBox** control.

UniqueID

ViewState

ViewStateIgnoresCase

Visible

Width

AddAttributesToRender

1  
2 [C#] protected override void AddAttributesToRender(HtmlTextWriter writer);

3 [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);

4 [VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As  
5 HtmlTextWriter)

6 [JScript] protected override function AddAttributesToRender(writer :  
7 HtmlTextWriter);

8  
9 *Description*

10 Adds name, size, multiple, and onchange to list of attributes to render. The  
11 output stream that renders HTML content to the client.

12 **OnPreRender**

13  
14 [C#] protected override void OnPreRender(EventArgs e);

15 [C++] protected: void OnPreRender(EventArgs\* e);

16 [VB] Overrides Protected Sub OnPreRender(ByVal e As EventArgs)

17 [JScript] protected override function OnPreRender(e : EventArgs);

18  
19 *Description*

20 **RenderContents**

21  
22 [C#] protected override void RenderContents(HtmlTextWriter writer);

23 [C++] protected: void RenderContents(HtmlTextWriter\* writer);

24 [VB] Overrides Protected Sub RenderContents(ByVal writer As HtmlTextWriter)

25 [JScript] protected override function RenderContents(writer : HtmlTextWriter);

1  
2 *Description*

3 IPostBackDataHandler.LoadPostData

4  
5 [C#] bool IPostBackDataHandler.LoadPostData(string postDataKey,  
6 NameValueCollection postCollection);

7 [C++] bool IPostBackDataHandler::LoadPostData(String\* postDataKey,  
8 NameValueCollection\* postCollection);

9 [VB] Function LoadPostData(ByVal postDataKey As String, ByVal  
10 postCollection As NameValueCollection) As Boolean Implements

11 IPostBackDataHandler.LoadPostData

12 [JScript] function IPostBackDataHandler.LoadPostData(postDataKey : String,  
13 postCollection : NameValueCollection) : Boolean;

14 IPostBackDataHandler.RaisePostDataChangedEvent

15  
16 [C#] void IPostBackDataHandler.RaisePostDataChangedEvent();

17 [C++] void IPostBackDataHandler::RaisePostDataChangedEvent();

18 [VB] Sub RaisePostDataChangedEvent() Implements

19 IPostBackDataHandler.RaisePostDataChangedEvent

20 [JScript] function IPostBackDataHandler.RaisePostDataChangedEvent();

21 ListControl class (System.Web.UI.WebControls)

22 TrackViewState

23  
24  
25 *Description*

Serves as the abstract base class that defines the properties, methods, and events common for all list-type controls.

The **System.Web.UI.WebControls.ListControl** class is typically not instantiated. Instead, to provide common basic functionality, this class is inherited by others, such as the **System.Web.UI.WebControls.CheckBoxList** , **System.Web.UI.WebControls.DropDownList** , **System.Web.UI.WebControls.ListBox** , and **System.Web.UI.WebControls.RadioButtonList** .

ListControl

*Example Syntax:*

TrackViewState

[C#] public ListControl();

[C++] public: ListControl();

[VB] Public Sub New()

[JScript] public function ListControl();

### *Description*

Initializes a new instance of the **System.Web.UI.WebControls.ListControl** class.

AccessKey

Attributes

AutoPostBack

TrackViewState

1  
2  
3 *Description*

4 Gets or sets a value indicating whether a postback to the server  
5 automatically occurs when the user changes the list selection.

6 Set this property to **true** if the server needs to capture the selection as soon  
7 as it is made. For example, other controls on the Web page can be automatically  
8 filled depending on the user's selection from a list control.

9 BackColor

10 BorderColor

11 BorderStyle

12 BorderWidth

13 ChildControlsCreated

14 ClientID

15 Context

16 Controls

17 ControlStyle

18 ControlStyleCreated

19 CssClass

20 DataMember

21 TrackViewState

22  
23  
24 *Description*  
25

1 Gets or sets the specific table in the  
2 **System.Web.UI.WebControls.ListControl.DataSource** to bind to the control.

3 If the **System.Web.UI.WebControls.ListControl.DataSource** contains  
4 more than one table, use this property to specify the exact table to bind to the  
5 control.

6 DataSource

7 ViewState

8  
9 [C#] public virtual object DataSource {get; set;}

10 [C++] public: \_\_property virtual Object\* get\_DataSource();public: \_\_property  
11 virtual void set\_DataSource(Object\*);

12 [VB] Overridable Public Property DataSource As Object

13 [JScript] public function get DataSource() : Object;public function set  
14 DataSource(Object);

15  
16 *Description*

17 Gets or sets the data source that populates the items of the list control.

18 Use this property to specify a source of data to populate a list control.

19 DataTextField

20 ViewState

21  
22 [C#] public virtual string DataTextField {get; set;}

23 [C++] public: \_\_property virtual String\* get\_DataTextField();public: \_\_property  
24 virtual void set\_DataTextField(String\*);

25 [VB] Overridable Public Property DataTextField As String

[JScript] public function get DataTextField() : String;public function set  
DataTextField(String);

#### *Description*

Gets or sets the field of the data source that provides the text content of the  
list items.

Use this property to specify a field in the  
**System.Web.UI.WebControls.ListControl.DataSource** to display as the items  
of the list in a list control.

DataTextFieldString

TrackViewState

[C#] public virtual string DataTextFieldString {get; set;}

[C++] public: \_\_property virtual String\* get\_DataTextFieldString();public:

\_\_property virtual void set\_DataTextFieldString(String\*);

[VB] Overridable Public Property DataTextFieldString As String

[JScript] public function get DataTextFieldString() : String;public function set  
DataTextFieldString(String);

#### *Description*

Gets or sets the formatting string used to control how data bound to the list  
control is displayed.

Use this property to provide a custom display format for the items in the list  
control.

DataValueField



1       TrackViewState

2

3   [C#] public virtual string DataValueField {get; set;}

4   [C++] public: \_\_property virtual String\* get\_DataValueField();public: \_\_property

5   virtual void set\_DataValueField(String\*);

6   [VB] Overridable Public Property DataValueField As String

7   [JScript] public function get DataValueField() : String;public function set

8   DataValueField(String);

9

10   *Description*

11       Gets or sets the field of the data source that provides the value of each list

12   item.

13       Use this property to specify the field that contains the value of each item in

14   a list control.

15       Enabled

16       EnableViewState

17       Events

18       Font

19       ForeColor

20       HasChildViewState

21       Height

22       ID

23       IsTrackingViewState

24       Items

25       TrackViewState

1  
2  
3 *Description*

4 Gets the collection of items in the list control.

5 Use this property to get the properties of items in the list control. This  
6 property can be used to determine the selected items in the list control.

7 NamingContainer

8 Page

9 Parent

10 SelectedIndex

11 TrackViewState

12  
13  
14 *Description*

15 Gets or sets the lowest ordinal index of the selected items in the list.

16 Use this property to determine the index of the currently selected item in  
17 the list if the list control allows only one selection. If the list control supports  
18 multiple selections, use this property to determine the lowest index of the selected  
19 items.

20 SelectedItem

21 TrackViewState

22  
23 [C#] public virtual ListItem SelectedItem {get;}

24 [C++] public: \_\_property virtual ListItem\* get\_SelectedItem();

25 [VB] Overridable Public ReadOnly Property SelectedItem As ListItem

1 [JScript] public function get SelectedItem() : ListItem;

2  
3 *Description*

4 Gets the selected item with the lowest index in the list control.

5 If the list control allows only a single selection, use this property to get the  
6 individual properties of the selected item. If the list control allows multiple  
7 selections, use this property to get the properties of the lowest indexed item  
8 selected from the list control.

9 Site

10 Style

11 TabIndex

12 TagKey

13 TagName

14 TemplateSourceDirectory

15 ToolTip

16 UniqueID

17 ViewState

18 ViewStateIgnoresCase

19 Visible

20 Width

21 TrackViewState

22  
23  
24 *Description*

Occurs when the selection on the list changes and is posted back to the server.

This event is raised when the selection from the list changes and is posted back to the server.

#### ClearSelection

[C#] public virtual void ClearSelection();

[C++] public: virtual void ClearSelection();

[VB] Overridable Public Sub ClearSelection()

[JScript] public function ClearSelection();

#### *Description*

Clears out the list selection and sets the **System.Web.UI.WebControls.ListItem.Selected** property of all items to false.

#### LoadViewState

[C#] protected override void LoadViewState(object savedState);

[C++] protected: void LoadViewState(Object\* savedState);

[VB] Overrides Protected Sub LoadViewState(ByVal savedState As Object)

[JScript] protected override function LoadViewState(savedState : Object);

#### *Description*

Load previously saved state. Overridden to restore selection.

#### OnDataBinding

1  
2 [C#] protected override void OnDataBinding(EventArgs e);  
3 [C++] protected: void OnDataBinding(EventArgs\* e);  
4 [VB] Overrides Protected Sub OnDataBinding(ByVal e As EventArgs)  
5 [JScript] protected override function OnDataBinding(e : EventArgs);  
6

7 *Description*

8       OnSelectedIndexChanged

9  
10 [C#] protected virtual void OnSelectedIndexChanged(EventArgs e);  
11 [C++] protected: virtual void OnSelectedIndexChanged(EventArgs\* e);  
12 [VB] Overridable Protected Sub OnSelectedIndexChanged(ByVal e As  
13 EventArgs)  
14 [JScript] protected function OnSelectedIndexChanged(e : EventArgs);  
15

16 *Description*

17       Raises the  
18 **System.Web.UI.WebControls.ListControl.SelectedIndexChanged** event.

19       Raising an event invokes the event handler through a delegate. For more  
20 information, see . An **System.EventArgs** that contains the event data.

21       SaveViewState

22  
23 [C#] protected override object SaveViewState();  
24 [C++] protected: Object\* SaveViewState();  
25 [VB] Overrides Protected Function SaveViewState() As Object

[JScript] protected override function SaveViewState() : Object;

*Description*

TrackViewState

[C#] protected override void TrackViewState();

[C++] protected: void TrackViewState();

[VB] Overrides Protected Sub TrackViewState()

[JScript] protected override function TrackViewState();

*Description*

ListItem class (System.Web.UI.WebControls)

TrackViewState

*Description*

Represents a data item in a databound list control. This class cannot be inherited.

A **System.Web.UI.WebControls.ListItem** control represents an individual data item within a databound list control, such as a **System.Web.UI.WebControls.ListBox** or a **System.Web.UI.WebControls.RadioButtonList** control.

ListItem

*Example Syntax:*

TrackViewState

1  
2 [C#] public ListItem();  
3 [C++] public: ListItem();  
4 [VB] Public Sub New()  
5 [JScript] public function ListItem(); Initializes a new instance of the  
6 **System.Web.UI.WebControls.ListItem** class.

7  
8 *Description*

9       Initializes a new instance of the **System.Web.UI.WebControls.ListItem**  
10 class.

11       ListItem

12       *Example Syntax:*

13       TrackViewState

14  
15 [C#] public ListItem(string text);  
16 [C++] public: ListItem(String\* text);  
17 [VB] Public Sub New(ByVal text As String)  
18 [JScript] public function ListItem(text : String);  
19

20 *Description*

21       Initializes a new instance of the **System.Web.UI.WebControls.ListItem**  
22 class with the specified text data. The text data to initialize the list item with.

23       ListItem

24       *Example Syntax:*

25       TrackViewState

```

1
2 [C#] public ListItem(string text, string value);
3 [C++] public: ListItem(String* text, String* value);
4 [VB] Public Sub New(ByVal text As String, ByVal value As String)
5 [JScript] public function ListItem(text : String, value : String);
6

```

### *Description*

Initializes a new instance of the **System.Web.UI.WebControls.ListItem** class with the specified text and value data.

The following table shows initial property values for an instance of **System.Web.UI.WebControls.ListItem** . The text data to initialize the list item with. The value data to initialize the list item with.

Attributes

TrackViewState

```

16 [C#] public AttributeCollection Attributes {get;}
17 [C++] public: __property AttributeCollection* get_Attributes();
18 [VB] Public ReadOnly Property Attributes As AttributeCollection
19 [JScript] public function get Attributes() : AttributeCollection;
20

```

### *Description*

Gets the collection of attribute name/value pairs expressed on the list item control but not supported by the control's strongly typed properties.

Selected

TrackViewState



1  
2 [C#] public bool Selected {get; set;}

3 [C++] public: \_\_property bool get\_Selected();public: \_\_property void  
4 set\_Selected(bool);

5 [VB] Public Property Selected As Boolean

6 [JScript] public function get Selected() : Boolean;public function set  
7 Selected(Boolean);

8  
9 *Description*

10 Specifies a value indicating whether the item is selected.

11 Text

12 TrackViewState

13  
14 [C#] public string Text {get; set;}

15 [C++] public: \_\_property String\* get\_Text();public: \_\_property void  
16 set\_Text(String\*);

17 [VB] Public Property Text As String

18 [JScript] public function get Text() : String;public function set Text(String);

19  
20 *Description*

21 Gets or sets the text displayed in the list control for the item represented by  
22 the **System.Web.UI.WebControls.ListItem** control.

23 If the list item control has no text content, then gets its value content, if  
24 exists, instead.

25 Value

## TrackViewState

[C#] public string Value {get; set;}

[C++] public: \_\_property String\* get\_Value();public: \_\_property void  
set\_Value(String\*);

[VB] Public Property Value As String

[JScript] public function get Value() : String;public function set Value(String);

### *Description*

Gets or sets the value content of the list item control.

If the list item control has no value content, then gets its text content, if  
exists, instead.

### *Equals*

[C#] public override bool Equals(object o);

[C++] public: bool Equals(Object\* o);

[VB] Overrides Public Function Equals(ByVal o As Object) As Boolean

[JScript] public override function Equals(o : Object) : Boolean;

### *Description*

### *FromString*

[C#] public static ListItem FromString(string s);

[C++] public: static ListItem\* FromString(String\* s);

[VB] Public Shared Function FromString(ByVal s As String) As ListItem

1 [JScript] public static function FromString(s : String) : ListItem;

2  
3 *Description*

4 Creates a **System.Web.UI.WebControls.ListItem** from the specified  
5 string. The specified string for creating a  
6 **System.Web.UI.WebControls.ListItem**.

7 **GetHashCode**

8  
9 [C#] public override int GetHashCode();

10 [C++] public: int GetHashCode();

11 [VB] Overrides Public Function GetHashCode() As Integer

12 [JScript] public override function GetHashCode() : int;

13  
14 *Description*

15  
16 **IAttributeAccessor.GetAttribute**

17  
18 [C#] string IAttributeAccessor.GetAttribute(string name);

19 [C++] String\* IAttributeAccessor::GetAttribute(String\* name);

20 [VB] Function GetAttribute(ByVal name As String) As String Implements  
21 IAttributeAccessor.GetAttribute

22 [JScript] function IAttributeAccessor.GetAttribute(name : String) : String;

23 **IAttributeAccessor.SetAttribute**

24  
25 [C#] void IAttributeAccessor.SetAttribute(string name, string value);

```

1 [C++] void IAttributeAccessor::SetAttribute(String* name, String* value);
2 [VB] Sub SetAttribute(ByVal name As String, ByVal value As String)
3 Implements IAttributeAccessor.SetAttribute
4 [JScript] function IAttributeAccessor.SetAttribute(name : String, value : String);
5     IParserAccessor.AddParsedSubObject
6
7 [C#] void IParserAccessor.AddParsedSubObject(object obj);
8 [C++] void IParserAccessor::AddParsedSubObject(Object* obj);
9 [VB] Sub AddParsedSubObject(ByVal obj As Object) Implements
10 IParserAccessor.AddParsedSubObject
11 [JScript] function IParserAccessor.AddParsedSubObject(obj : Object);
12     IStateManager.LoadViewState
13
14 [C#] void IStateManager.LoadViewState(object state);
15 [C++] void IStateManager::LoadViewState(Object* state);
16 [VB] Sub LoadViewState(ByVal state As Object) Implements
17 IStateManager.LoadViewState
18 [JScript] function IStateManager.LoadViewState(state : Object);
19     IStateManager.SaveViewState
20
21 [C#] object IStateManager.SaveViewState();
22 [C++] Object* IStateManager::SaveViewState();
23 [VB] Function SaveViewState() As Object Implements
24 IStateManager.SaveViewState
25 [JScript] function IStateManager.SaveViewState() : Object;

```

1            IStateManager.TrackViewState

2

3    [C#] void IStateManager.TrackViewState();

4    [C++] void IStateManager::TrackViewState();

5    [VB] Sub TrackViewState() Implements IStateManager.TrackViewState

6    [JScript] function IStateManager.TrackViewState();

7            ToString

8

9    [C#] public override string ToString();

10    [C++] public: String\* ToString();

11    [VB] Overrides Public Function ToString() As String

12    [JScript] public override function ToString() : String;

13

14    *Description*

15            ListItemCollection class (System.Web.UI.WebControls)

16            ToString

17

18

19    *Description*

20            Encapsulates the **System.Web.UI.WebControls.ListItem** controls within

21    a **System.Web.UI.WebControls.ListControl** . This class cannot be inherited.

22            ListItemCollection

23    *Example Syntax:*

24            ToString

25

```

1
2 [C#] public ListItemCollection();
3 [C++] public: ListItemCollection();
4 [VB] Public Sub New()
5 [JScript] public function ListItemCollection();
6

```

### *Description*

Initializes a new instance of the  
**System.Web.UI.WebControls.ListItemCollection** class.

Capacity

ToString

```

13 [C#] public int Capacity {get; set;}
14 [C++] public: __property int get_Capacity();public: __property void
15 set_Capacity(int);
16 [VB] Public Property Capacity As Integer
17 [JScript] public function get Capacity() : int;public function set Capacity(int);

```

Count

ToString

```

21 [C#] public int Count {get;}
22 [C++] public: __property int get_Count();
23 [VB] Public ReadOnly Property Count As Integer
24 [JScript] public function get Count() : int;

```

1  
2 *Description*

3 Gets the item count of the collection.

4 IsReadOnly

5 ToString

6  
7 [C#] public bool IsReadOnly {get;}

8 [C++] public: \_\_property bool get\_IsReadOnly();

9 [VB] Public ReadOnly Property IsReadOnly As Boolean

10 [JScript] public function get IsReadOnly() : Boolean;

11  
12 *Description*

13 Gets a value indicating whether the collection is read-only.

14 IsSynchronized

15 ToString

16  
17 [C#] public bool IsSynchronized {get;}

18 [C++] public: \_\_property bool get\_IsSynchronized();

19 [VB] Public ReadOnly Property IsSynchronized As Boolean

20 [JScript] public function get IsSynchronized() : Boolean;

21  
22 *Description*

23 Gets a value indicating whether access to the collection is synchronized  
24 (thread-safe).

25 Item

ToString

[C#] public ListItem this[int index] {get;}

[C++] public: \_\_property ListItem\* get\_Item(int index);

[VB] Public Default ReadOnly Property Item(ByVal index As Integer) As  
ListItem

[JScript] returnValue = ListItemCollectionObject.Item(index);

*Description*

Gets a **System.Web.UI.WebControls.ListItem** referenced by the specified ordinal index value. An ordinal index value that specifies which **System.Web.UI.WebControls.ListItem** to return.

SyncRoot

ToString

[C#] public object SyncRoot {get;}

[C++] public: \_\_property Object\* get\_SyncRoot();

[VB] Public ReadOnly Property SyncRoot As Object

[JScript] public function get SyncRoot() : Object;

*Description*

Gets the object that can be used to synchronize access to the collection. In this case, it is the collection itself.

Add



```

1
2 [C#] public void Add(ListItem item);
3 [C++] public: void Add(ListItem* item);
4 [VB] Public Sub Add(ByVal item As ListItem)
5 [JScript] public function Add(item : ListItem);
6

```

### *Description*

Adds the specified **System.Web.UI.WebControls.ListItem** to the end of the collection. The **System.Web.UI.WebControls.ListItem** to add to the collection.

#### Add

```

12
13 [C#] public void Add(string item);
14 [C++] public: void Add(String* item);
15 [VB] Public Sub Add(ByVal item As String)
16 [JScript] public function Add(item : String); Adds the specified item to the end of
17 the collection.
18

```

### *Description*

Adds the specified item to the end of the collection. A **System.String** that specifies the item to add.

#### AddRange

```

23
24 [C#] public void AddRange(ListItem[] items);
25 [C++] public: void AddRange(ListItem* items[]);

```

1 [VB] Public Sub AddRange(ByVal items() As ListItem)

2 [JScript] public function AddRange(items : ListItem[]);

3 Clear

4

5 [C#] public void Clear();

6 [C++] public: \_\_sealed void Clear();

7 [VB] NotOverridable Public Sub Clear()

8 [JScript] public function Clear();

9

10 *Description*

11 Removes all **System.Web.UI.WebControls.ListItem** controls from the  
12 collection.

13 Contains

14

15 [C#] public bool Contains(ListItem item);

16 [C++] public: bool Contains(ListItem\* item);

17 [VB] Public Function Contains(ByVal item As ListItem) As Boolean

18 [JScript] public function Contains(item : ListItem) : Boolean;

19

20 *Description*

21 Returns a value indicating whether the collection contains the specified  
22 item.

23 *Return Value:* **true** if the collection contains the specified item; otherwise, **false** .

24 A **System.Web.UI.WebControls.ListItem** to search for in the collection.

25 CopyTo

```

1
2 [C#] public void CopyTo(Array array, int index);
3 [C++] public: __sealed void CopyTo(Array* array, int index);
4 [VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As
5 Integer)
6 [JScript] public function CopyTo(array : Array, index : int);
7

```

### *Description*

Copies contents from the collection to a specified **System.Array** with a specified starting index. The specified **System.Array** that receives copied contents from the collection. The starting position in the specified **System.Array** to receive copied contents.

### **FindByText**

```

13
14
15 [C#] public ListItem FindByText(string text);
16 [C++] public: ListItem* FindByText(String* text);
17 [VB] Public Function FindByText(ByVal text As String) As ListItem
18 [JScript] public function FindByText(text : String) : ListItem;

```

### **FindByValue**

```

19
20
21 [C#] public ListItem FindByValue(string value);
22 [C++] public: ListItem* FindByValue(String* value);
23 [VB] Public Function FindByValue(ByVal value As String) As ListItem
24 [JScript] public function FindByValue(value : String) : ListItem;

```

### **GetEnumerator**

```

1
2 [C#] public IEnumerator GetEnumerator();
3 [C++] public: __sealed IEnumerator* GetEnumerator();
4 [VB] NotOverridable Public Function GetEnumerator() As IEnumerator
5 [JScript] public function GetEnumerator() : IEnumerator;
6

```

### *Description*

Returns an enumerator of all **System.Web.UI.WebControls.ListItem** controls within the collection.

*Return Value:* An enumerator that enumerates over all **System.Web.UI.WebControls.ListItem** controls within the collection.

### IndexOf

```

14 [C#] public int IndexOf(ListItem item);
15 [C++] public: int IndexOf(ListItem* item);
16 [VB] Public Function IndexOf(ByVal item As ListItem) As Integer
17 [JScript] public function IndexOf(item : ListItem) : int;
18

```

### *Description*

Returns an ordinal index value that represents the position of the specified **System.Web.UI.WebControls.ListItem** within the collection.

*Return Value:* The ordinal index position of the specified **System.Web.UI.WebControls.ListItem** within the collection. The specified **System.Web.UI.WebControls.ListItem** to search for in the collection.

### Insert

1  
2 [C#] public void Insert(int index, ListItem item);

3 [C++] public: void Insert(int index, ListItem\* item);

4 [VB] Public Sub Insert(ByVal index As Integer, ByVal item As ListItem)

5 [JScript] public function Insert(index : int, item : ListItem);

6  
7 *Description*

8 Inserts the specified **System.Web.UI.WebControls.ListItem** to the  
9 collection at the specified index location. The location in the collection to add the  
10 **System.Web.UI.WebControls.ListItem** . The  
11 **System.Web.UI.WebControls.ListItem** to add to the collection.

12 Insert

13  
14 [C#] public void Insert(int index, string item);

15 [C++] public: void Insert(int index, String\* item);

16 [VB] Public Sub Insert(ByVal index As Integer, ByVal item As String)

17 [JScript] public function Insert(index : int, item : String); Inserts the specified item  
18 to the collection at the specified index location.

19  
20 *Description*

21 Adds the specified item to the collection at the specified index location.  
22 The location in the collection to add the **System.Web.UI.WebControls.ListItem** .  
23 The item to add to the collection.

24 Remove

```

1
2 [C#] public void Remove(ListItem item);
3 [C++] public: void Remove(ListItem* item);
4 [VB] Public Sub Remove(ByVal item As ListItem)
5 [JScript] public function Remove(item : ListItem);
6

```

#### *Description*

Removes the specified **System.Web.UI.WebControls.ListItem** from the collection. The **System.Web.UI.WebControls.ListItem** to remove from the collection.

#### **Remove**

```

13 [C#] public void Remove(string item);
14 [C++] public: void Remove(String* item);
15 [VB] Public Sub Remove(ByVal item As String)
16 [JScript] public function Remove(item : String); Removes the specified item from
17 the collection.
18

```

#### *Description*

Removes the specified item from the collection. The item to remove from the collection.

#### **RemoveAt**

```

24 [C#] public void RemoveAt(int index);
25 [C++] public: __sealed void RemoveAt(int index);

```

1 [VB] NotOverridable Public Sub RemoveAt(ByVal index As Integer)

2 [JScript] public function RemoveAt(index : int);

3  
4 *Description*

5 Removes the **System.Web.UI.WebControls.ListItem** from the collection  
6 at the specified index location. The location in the collection to remove the  
7 **System.Web.UI.WebControls.ListItem**.

8 **IList.Add**

9  
10 [C#] int IList.Add(object item);

11 [C++] int IList::Add(Object\* item);

12 [VB] Function Add(ByVal item As Object) As Integer Implements IList.Add

13 [JScript] function IList.Add(item : Object) : int;

14 **IList.Contains**

15  
16 [C#] bool IList.Contains(object item);

17 [C++] bool IList::Contains(Object\* item);

18 [VB] Function Contains(ByVal item As Object) As Boolean Implements

19 **IList.Contains**

20 [JScript] function IList.Contains(item : Object) : Boolean;

21 **IList.IndexOf**

22  
23 [C#] int IList.IndexOf(object item);

24 [C++] int IList::IndexOf(Object\* item);

25 [VB] Function IndexOf(ByVal item As Object) As Integer Implements

```

1  IList.IndexOf
2  [JScript] function IList.IndexOf(item : Object) : int;
3
4      IList.Insert
5
6  [C#] void IList.Insert(int index, object item);
7  [C++] void IList::Insert(int index, Object* item);
8  [VB] Sub Insert(ByVal index As Integer, ByVal item As Object) Implements
9      IList.Insert
10 [JScript] function IList.Insert(index : int, item : Object);
11
12      IList.Remove
13
14 [C#] void IList.Remove(object item);
15 [C++] void IList::Remove(Object* item);
16 [VB] Sub Remove(ByVal item As Object) Implements IList.Remove
17 [JScript] function IList.Remove(item : Object);
18
19      IStateManager.LoadViewState
20
21 [C#] void IStateManager.LoadViewState(object state);
22 [C++] void IStateManager::LoadViewState(Object* state);
23 [VB] Sub LoadViewState(ByVal state As Object) Implements
24      IStateManager.LoadViewState
25 [JScript] function IStateManager.LoadViewState(state : Object);
26
27      IStateManager.SaveViewState
28
29 [C#] object IStateManager.SaveViewState();

```



1 [C++] Object\* IStateManager::SaveViewState();  
2 [VB] Function SaveViewState() As Object Implements  
3 IStateManager.SaveViewState  
4 [JScript] function IStateManager.SaveViewState() : Object;  
5 IStateManager.TrackViewState  
6  
7 [C#] void IStateManager.TrackViewState();  
8 [C++] void IStateManager::TrackViewState();  
9 [VB] Sub TrackViewState() Implements IStateManager.TrackViewState  
10 [JScript] function IStateManager.TrackViewState();  
11 ListItemControlBuilder class (System.Web.UI.WebControls)  
12 ToString

### *Description*

Interacts with the parser to build a  
**System.Web.UI.WebControls.ListItem** control.

ListItemControlBuilder

### *Example Syntax:*

ToString

22 [C#] public ListItemControlBuilder();  
23 [C++] public: ListItemControlBuilder();  
24 [VB] Public Sub New()  
25 [JScript] public function ListItemControlBuilder();

1	ControlType
2	FChildrenAsProperties
3	FIsNonParserAccessor
4	HasAspCode
5	ID
6	InDesigner
7	NamingContainerType
8	Parser
9	TagName
10	AllowWhitespaceLiterals
11	
12	[C#] public override bool AllowWhitespaceLiterals();
13	[C++] public: bool AllowWhitespaceLiterals();
14	[VB] Overrides Public Function AllowWhitespaceLiterals() As Boolean
15	[JScript] public override function AllowWhitespaceLiterals() : Boolean;
16	
17	<i>Description</i>
18	
19	HtmlDecodeLiterals
20	
21	[C#] public override bool HtmlDecodeLiterals();
22	[C++] public: bool HtmlDecodeLiterals();
23	[VB] Overrides Public Function HtmlDecodeLiterals() As Boolean
24	[JScript] public override function HtmlDecodeLiterals() : Boolean;
25	

1  
2 *Description*

3  
4 ListItemType enumeration (System.Web.UI.WebControls)

5 ToString  
6  
7

8 *Description*

9 Specifies the type of an item in a list control.

10 The **System.Web.UI.WebControls.ListItemType** enumeration represents  
11 the different items that can be included in a list control, such as  
12 **System.Web.UI.WebControls.DataGrid** ,  
13 **System.Web.UI.WebControls.DataList** , and the  
14 **System.Web.UI.WebControls.Repeater** . A typical list control consists of cells  
15 that contain elements represented by this enumeration.

16 ToString  
17

18 [C#] public const ListItemType AlternatingItem;

19 [C++] public: const ListItemType AlternatingItem;

20 [VB] Public Const AlternatingItem As ListItemType

21 [JScript] public var AlternatingItem : ListItemType;

22  
23 *Description*

24 An item in alternating (zero-based even-indexed) cells. It is databound.

25 ToString

1  
2 [C#] public const ListItemType EditItem;  
3 [C++] public: const ListItemType EditItem;  
4 [VB] Public Const EditItem As ListItemType  
5 [JScript] public var EditItem : ListItemType;  
6

7 *Description*

8 An item in a list control currently in edit mode. It is databound.

9 ToString  
10

11 [C#] public const ListItemType Footer;  
12 [C++] public: const ListItemType Footer;  
13 [VB] Public Const Footer As ListItemType  
14 [JScript] public var Footer : ListItemType;  
15

16 *Description*

17 A footer for the list control. It is not databound.

18 ToString  
19

20 [C#] public const ListItemType Header;  
21 [C++] public: const ListItemType Header;  
22 [VB] Public Const Header As ListItemType  
23 [JScript] public var Header : ListItemType;  
24

25 *Description*

A header for the list control. It is not databound.

ToString

[C#] public const ListItemType Item;

[C++] public: const ListItemType Item;

[VB] Public Const Item As ListItemType

[JScript] public var Item : ListItemType;

*Description*

An item in the list control. It is databound.

ToString

[C#] public const ListItemType Pager;

[C++] public: const ListItemType Pager;

[VB] Public Const Pager As ListItemType

[JScript] public var Pager : ListItemType;

*Description*

A pager that displays the controls to navigate to different pages associated with the **System.Web.UI.WebControls.DataGrid** control. It is not databound.

ToString

[C#] public const ListItemType SelectedItem;

[C++] public: const ListItemType SelectedItem;

[VB] Public Const SelectedItem As ListItemType

1 [JScript] public var SelectedItem : ListItemType;

3 *Description*

4 A selected item in the list control. It is databound.

5 ToString

7 [C#] public const ListItemType Separator;

8 [C++] public: const ListItemType Separator;

9 [VB] Public Const Separator As ListItemType

10 [JScript] public var Separator : ListItemType;

12 *Description*

13 A separator between items in a list control. It is not databound.

14 ListSelectionMode enumeration (System.Web.UI.WebControls)

15 ToString

18 *Description*

19 Specifies the selection mode of the **System.Web.UI.WebControls.ListBox**  
20 control.

21 The **System.Web.UI.WebControls.ListSelectionMode** enumeration  
22 represents the selection mode of the **System.Web.UI.WebControls.ListBox**  
23 control that determines whether a user can select multiple items or just a single  
24 item.

25 ToString

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#] public const ListSelectionMode Multiple;  
[C++] public: const ListSelectionMode Multiple;  
[VB] Public Const Multiple As ListSelectionMode  
[JScript] public var Multiple : ListSelectionMode;
```

*Description*

Multiple item selection mode.  
ToString

```
[C#] public const ListSelectionMode Single;  
[C++] public: const ListSelectionMode Single;  
[VB] Public Const Single As ListSelectionMode  
[JScript] public var Single : ListSelectionMode;
```

*Description*

Single item selection mode.  
Literal class (System.Web.UI.WebControls)  
ToString

*Description*

Literal  
*Example Syntax:*

1	ToString
2	
3	[C#] public Literal();
4	[C++] public: Literal();
5	[VB] Public Sub New()
6	[JScript] public function Literal();
7	ChildControlsCreated
8	ClientID
9	Context
10	Controls
11	EnableViewState
12	Events
13	HasChildViewState
14	ID
15	IsTrackingViewState
16	NamingContainer
17	Page
18	Parent
19	Site
20	TemplateSourceDirectory
21	Text
22	ToString
23	UniqueID
24	ViewState
25	ViewStateIgnoresCase

FOR RELEASE



1 Visible

2 AddParsedSubObject

3

4 [C#] protected override void AddParsedSubObject(object obj);

5 [C++] protected: void AddParsedSubObject(Object\* obj);

6 [VB] Overrides Protected Sub AddParsedSubObject(ByVal obj As Object)

7 [JScript] protected override function AddParsedSubObject(obj : Object);

8 CreateControlCollection

9

10 [C#] protected override ControlCollection CreateControlCollection();

11 [C++] protected: ControlCollection\* CreateControlCollection();

12 [VB] Overrides Protected Function CreateControlCollection() As

13 ControlCollection

14 [JScript] protected override function CreateControlCollection() :

15 ControlCollection;

16 Render

17

18 [C#] protected override void Render(HtmlTextWriter output);

19 [C++] protected: void Render(HtmlTextWriter\* output);

20 [VB] Overrides Protected Sub Render(ByVal output As HtmlTextWriter)

21 [JScript] protected override function Render(output : HtmlTextWriter);

22 LiteralControlBuilder class (System.Web.UI.WebControls)

23 TrackViewState

24 LiteralControlBuilder

25 *Example Syntax:*

1 TrackViewState  
2 ControlType  
3 FChildrenAsProperties  
4 FIsNonParserAccessor  
5 HasAspCode  
6 ID  
7 InDesigner  
8 NamingContainerType  
9 Parser  
10 TagName  
11 AllowWhitespaceLiterals

12  
13 [C#] public override bool AllowWhitespaceLiterals();  
14 [C++] public: bool AllowWhitespaceLiterals();  
15 [VB] Overrides Public Function AllowWhitespaceLiterals() As Boolean  
16 [JScript] public override function AllowWhitespaceLiterals() : Boolean;

17 AppendSubBuilder

18  
19 [C#] public override void AppendSubBuilder(ControlBuilder subBuilder);  
20 [C++] public: void AppendSubBuilder(ControlBuilder\* subBuilder);  
21 [VB] Overrides Public Sub AppendSubBuilder(ByVal subBuilder As  
22 ControlBuilder)  
23 [JScript] public override function AppendSubBuilder(subBuilder :  
24 ControlBuilder);

25 MonthChangedEventArgs class (System.Web.UI.WebControls)

ToString

*Description*

Provides data for the **System.Web.UI.WebControls.Calendar.VisibleMonthChanged** event of a **System.Web.UI.WebControls.Calendar** . This class cannot be inherited.

The **System.Web.UI.WebControls.Calendar.VisibleMonthChanged** event is raised when the user clicks on the **System.Web.UI.WebControls.Calendar** navigation controls to display the next or previous month.

MonthChangedEventArgs

*Example Syntax:*

ToString

[C#] public MonthChangedEventArgs(DateTime newDate, DateTime  
previousDate);

[C++] public: MonthChangedEventArgs(DateTime newDate, DateTime  
previousDate);

[VB] Public Sub New(ByVal newDate As DateTime, ByVal previousDate As  
DateTime)

[JScript] public function MonthChangedEventArgs(newDate : DateTime,  
previousDate : DateTime);

*Description*

1        Initializes a new instance of the  
2        **System.Web.UI.WebControls.MonthChangedEventArgs** class.

3        The following table shows initial property values for an instance of  
4        **System.Web.UI.WebControls.MonthChangedEventArgs** . The date that  
5        determines the month currently displayed by the  
6        **System.Web.UI.WebControls.Calendar**. The date that determines the month  
7        previously displayed by the **System.Web.UI.WebControls.Calendar**.

8        **NewDate**

9        **ToString**

10  
11        [C#] public DateTime NewDate {get;}

12        [C++] public: \_\_property DateTime get\_NewDate();

13        [VB] Public ReadOnly Property NewDate As DateTime

14        [JScript] public function get NewDate() : DateTime;

15  
16        *Description*

17        Gets the date that determines the currently displayed month in the  
18        **System.Web.UI.WebControls.Calendar** .

19        Use this property to get the month currently displayed by the  
20        **System.Web.UI.WebControls.Calendar** and to compare the currently displayed  
21        month with another month. For example, you can use this property to calculate the  
22        number of months until or since a certain date.

23        **PreviousDate**

24        **ToString**

```

1
2 [C#] public DateTime PreviousDate {get;}
3 [C++] public: __property DateTime get_PreviousDate();
4 [VB] Public ReadOnly Property PreviousDate As DateTime
5 [JScript] public function get PreviousDate() : DateTime;
6

```

### 7 *Description*

8 Gets the date that determined the previously displayed month in the  
9 **System.Web.UI.WebControls.Calendar** .

10 Use this property to determine the previously displayed month on the  
11 **System.Web.UI.WebControls.Calendar** . This date can be used to restore the  
12 **System.Web.UI.WebControls.Calendar** back to the previously displayed month  
13 by setting the **System.Web.UI.WebControls.Calendar.VisibleDate** property to  
14 **System.Web.UI.WebControls.MonthChangedEventArgs.PreviousDate** .

15 MonthChangedEventHandler delegate (System.Web.UI.WebControls)

16 ToString

### 17 18 19 *Description*

20 Represents the method that handles the  
21 **System.Web.UI.WebControls.Calendar.VisibleMonthChanged** event of a  
22 **System.Web.UI.WebControls.Calendar** . The source of the event. A  
23 **System.Web.UI.WebControls.MonthChangedEventArgs** that contains the  
24 event data.

The **System.Web.UI.WebControls.Calendar.VisibleMonthChanged** event is raised when the user clicks on the navigation controls for the **System.Web.UI.WebControls.Calendar** to display the next or previous month.

NextPrevFormat enumeration (System.Web.UI.WebControls)  
ToString

#### *Description*

Represents the display format for the previous and next month navigation controls within the **System.Web.UI.WebControls.Calendar** .

The **System.Web.UI.WebControls.NextPrevFormat** enumeration represents the different styles for the next and previous month buttons on the **System.Web.UI.WebControls.Calendar** .

ToString

[C#] public const NextPrevFormat CustomText;

[C++] public: const NextPrevFormat CustomText;

[VB] Public Const CustomText As NextPrevFormat

[JScript] public var CustomText : NextPrevFormat;

#### *Description*

Custom text format for the month navigation controls on the **System.Web.UI.WebControls.Calendar** .

ToString

[C#] public const NextPrevFormat FullMonth;  
[C++] public: const NextPrevFormat FullMonth;  
[VB] Public Const FullMonth As NextPrevFormat  
[JScript] public var FullMonth : NextPrevFormat;

*Description*

Full month name format for the month navigation controls on the  
**System.Web.UI.WebControls.Calendar** . For example, "January".

ToString

[C#] public const NextPrevFormat ShortMonth;  
[C++] public: const NextPrevFormat ShortMonth;  
[VB] Public Const ShortMonth As NextPrevFormat  
[JScript] public var ShortMonth : NextPrevFormat;

*Description*

Abbreviated month name format for the month navigation controls on the  
**System.Web.UI.WebControls.Calendar** . For example, "Jan".

PagedDataSource class (System.Web.UI.WebControls)

ToString

*Description*

Provides a wrapper over an **System.Collections.ICollection** data source to implement paging semantics or 'paged views' on top of the underlying datasource.

This class cannot be inherited.

This class uses the best available interface to enumerate over the data belonging to the current page. If the underlying data source supports indexed access (like **System.Array** and **System.Collections.IList** ), this wrapper uses it. Otherwise it falls back on **System.Collections.IEnumerable** .

PagedDataSource

*Example Syntax:*

ToString

[C#] public PagedDataSource();

[C++] public: PagedDataSource();

[VB] Public Sub New()

[JScript] public function PagedDataSource();

*Description*

Initializes a new instance of the **System.Web.UI.WebControls.PagedDataSource** class.

The following table shows initial property values for an instance of **System.Web.UI.WebControls.PagedDataSource** .

AllowCustomPaging

ToString

[C#] public bool AllowCustomPaging {get; set;}



1 [C++] public: \_\_property bool get \_AllowCustomPaging();public: \_\_property void  
2 set \_AllowCustomPaging(bool);

3 [VB] Public Property AllowCustomPaging As Boolean

4 [JScript] public function get AllowCustomPaging() : Boolean;public function set  
5 AllowCustomPaging(Boolean);

6  
7 *Description*

8 Gets or sets a value indicating whether to assume the underlying data  
9 source contains data for just the current page.

10 This is only applicable if AllowPaging is true. In this scenario, the  
11 developer can fetch just the right number of items from the DataSet to fill up a  
12 single page, which may be more efficient based on the source of the data.

13 AllowPaging

14 ToString

15  
16 [C#] public bool AllowPaging {get; set;}

17 [C++] public: \_\_property bool get \_AllowPaging();public: \_\_property void  
18 set \_AllowPaging(bool);

19 [VB] Public Property AllowPaging As Boolean

20 [JScript] public function get AllowPaging() : Boolean;public function set  
21 AllowPaging(Boolean);

22  
23 *Description*

24 Gets or sets a value indicating whether to implement page semantics on top  
25 of the underlying datasource.

1 Count  
2 ToString  
3  
4 [C#] public int Count {get;}  
5 [C++] public: \_\_property int get\_Count();  
6 [VB] Public ReadOnly Property Count As Integer  
7 [JScript] public function get Count() : int;

8  
9 *Description*

10 Gets the number of items to be used from the datasource.  
11 This takes into account several factors such as paged/non-paged modes,  
12 custom paging, and last page.

13 CurrentPageIndex  
14 ToString  
15

16 [C#] public int CurrentPageIndex {get; set;}  
17 [C++] public: \_\_property int get\_CurrentPageIndex();public: \_\_property void  
18 set\_CurrentPageIndex(int);  
19 [VB] Public Property CurrentPageIndex As Integer  
20 [JScript] public function get CurrentPageIndex() : int;public function set  
21 CurrentPageIndex(int);  
22

23 *Description*

24 Gets or sets a value indicating the index of the current page.

25 DataSource

ToString

[C#] public IEnumerable DataSource {get; set;}

[C++] public: \_\_property IEnumerable\* get\_DataSource();public: \_\_property void

set\_DataSource(IEnumerable\*);

[VB] Public Property DataSource As IEnumerable

[JScript] public function get DataSource() : IEnumerable;public function set

DataSource(IEnumerable);

#### *Description*

Gets or sets the data source.

DataSourceCount

ToString

[C#] public int DataSourceCount {get;}

[C++] public: \_\_property int get\_DataSourceCount();

[VB] Public ReadOnly Property DataSourceCount As Integer

[JScript] public function get DataSourceCount() : int;

#### *Description*

Gets the item count of the data source.

FirstIndexInPage

ToString

[C#] public int FirstIndexInPage {get;}

1 [C++] public: \_\_property int get\_FirstIndexInPage();

2 [VB] Public ReadOnly Property FirstIndexInPage As Integer

3 [JScript] public function get FirstIndexInPage() : int;

4  
5 *Description*

6 Gets the first index in the page.

7 IsCustomPagingEnabled

8 ToString

9  
10 [C#] public bool IsCustomPagingEnabled {get;}

11 [C++] public: \_\_property bool get\_IsCustomPagingEnabled();

12 [VB] Public ReadOnly Property IsCustomPagingEnabled As Boolean

13 [JScript] public function get IsCustomPagingEnabled() : Boolean;

14  
15 *Description*

16 Gets a value indicating whether custom paging is enabled.

17 IsFirstPage

18 ToString

19  
20 [C#] public bool IsFirstPage {get;}

21 [C++] public: \_\_property bool get\_IsFirstPage();

22 [VB] Public ReadOnly Property IsFirstPage As Boolean

23 [JScript] public function get IsFirstPage() : Boolean;

24  
25 *Description*

1 Gets a value indicating if the current page is the first page.

2 IsLastPage

3 ToString

4

5 [C#] public bool IsLastPage {get;}

6 [C++] public: \_\_property bool get\_IsLastPage();

7 [VB] Public ReadOnly Property IsLastPage As Boolean

8 [JScript] public function get IsLastPage() : Boolean;

9

10 *Description*

11 Gets a value indicating if the current page is the last page.

12 IsPagingEnabled

13 ToString

14

15 [C#] public bool IsPagingEnabled {get;}

16 [C++] public: \_\_property bool get\_IsPagingEnabled();

17 [VB] Public ReadOnly Property IsPagingEnabled As Boolean

18 [JScript] public function get IsPagingEnabled() : Boolean;

19

20 *Description*

21 Gets a value indicating whether paging is enabled.

22 IsReadOnly

23 ToString

24

25 [C#] public bool IsReadOnly {get;}

PDF-00000000

1 [C++] public: \_\_property bool get\_IsReadOnly();

2 [VB] Public ReadOnly Property IsReadOnly As Boolean

3 [JScript] public function get IsReadOnly() : Boolean;

4  
5 *Description*

6 Gets a value indicating whether the data source is read-only.

7 IsSynchronized

8 ToString

9  
10 [C#] public bool IsSynchronized {get;}

11 [C++] public: \_\_property bool get\_IsSynchronized();

12 [VB] Public ReadOnly Property IsSynchronized As Boolean

13 [JScript] public function get IsSynchronized() : Boolean;

14  
15 *Description*

16 Gets a value indicating whether access to the data source is synchronized  
17 (thread-safe).

18 PageCount

19 ToString

20  
21 [C#] public int PageCount {get;}

22 [C++] public: \_\_property int get\_PageCount();

23 [VB] Public ReadOnly Property PageCount As Integer

24 [JScript] public function get PageCount() : int;

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

*Description*

Gets the page count.

PageSize

ToString

```
[C#] public int PageSize {get; set;}
[C++] public: __property int get_PageSize();public: __property void
set_PageSize(int);
[VB] Public Property PageSize As Integer
[JScript] public function get PageSize() : int;public function set PageSize(int);
```

*Description*

Gets or sets the page size.

SyncRoot

ToString

```
[C#] public object SyncRoot {get;}
[C++] public: __property Object* get_SyncRoot();
[VB] Public ReadOnly Property SyncRoot As Object
[JScript] public function get SyncRoot() : Object;
```

*Description*

Gets the object that can be used to synchronize access to the collection. In this case, it is the collection itself.

VirtualCount

ToString

[C#] public int VirtualCount {get; set;}

[C++] public: \_\_property int get\_VirtualCount();public: \_\_property void

set\_VirtualCount(int);

[VB] Public Property VirtualCount As Integer

[JScript] public function get VirtualCount() : int;public function set

VirtualCount(int);

### *Description*

Gets or sets the virtual count.

CopyTo

[C#] public void CopyTo(Array array, int index);

[C++] public: \_\_sealed void CopyTo(Array\* array, int index);

[VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As

Integer)

[JScript] public function CopyTo(array : Array, index : int);

### *Description*

Copies contents from the data source to the specified **System.Array** with the specified starting index. The specified **System.Array** that receives copied contents from the data source. The starting position in the specified **System.Array** to receive copied contents.



## GetEnumerator

```
[C#] public IEnumerator GetEnumerator();  
[C++] public: __sealed IEnumerator* GetEnumerator();  
[VB] NotOverridable Public Function GetEnumerator() As IEnumerator  
[JScript] public function GetEnumerator() : IEnumerator;
```

### *Description*

Returns an enumerator of all items within the data source.

*Return Value:* An enumerator that enumerates over all items within the data source.

## GetItemProperties

```
[C#] public PropertyDescriptorCollection GetItemProperties(PropertyDescriptor[]  
listAccessors);  
[C++] public: __sealed PropertyDescriptorCollection*  
GetItemProperties(PropertyDescriptor* listAccessors[]);  
[VB] NotOverridable Public Function GetItemProperties(ByVal listAccessors()  
As PropertyDescriptor) As PropertyDescriptorCollection  
[JScript] public function GetItemProperties(listAccessors : PropertyDescriptor[]) :  
PropertyDescriptorCollection;
```

## GetListName

```
[C#] public string GetListName(PropertyDescriptor[] listAccessors);  
[C++] public: __sealed String* GetListName(PropertyDescriptor*
```

listAccessors[]);

[VB] NotOverridable Public Function GetListName(ByVal listAccessors() As

PropertyDescriptor) As String

[JScript] public function GetListName(listAccessors : PropertyDescriptor[]) :

String;

PagerMode enumeration (System.Web.UI.WebControls)

ToString

### *Description*

Represents the mode of the pager for accessing various pages within the **System.Web.UI.WebControls.DataGrid** control.

The **System.Web.UI.WebControls.PagerMode** enumeration represents the different display modes for the pager of a **System.Web.UI.WebControls.DataGrid** control.

ToString

[C#] public const PagerMode NextPrev;

[C++] public: const PagerMode NextPrev;

[VB] Public Const NextPrev As PagerMode

[JScript] public var NextPrev : PagerMode;

### *Description*

A pager with **Previous** and **Next** buttons to access the next and previous pages.

1 ToString  
2  
3 [C#] public const PagerMode NumericPages;  
4 [C++] public: const PagerMode NumericPages;  
5 [VB] Public Const NumericPages As PagerMode  
6 [JScript] public var NumericPages : PagerMode;  
7

8 *Description*

9 A pager with numbered buttons to access pages directly.

10 PagerPosition enumeration (System.Web.UI.WebControls)

11 ToString  
12  
13

14 *Description*

15 Specifies the position of the pager for accessing various pages within the  
16 **System.Web.UI.WebControls.DataGrid** control.

17 The **System.Web.UI.WebControls.PagerPosition** enumeration represents  
18 the different locations where the pager can be displayed.

19 ToString  
20

21 [C#] public const PagerPosition Bottom;  
22 [C++] public: const PagerPosition Bottom;  
23 [VB] Public Const Bottom As PagerPosition  
24 [JScript] public var Bottom : PagerPosition;  
25

1  
2 *Description*

3 A pager positioned at the bottom of the  
4 **System.Web.UI.WebControls.DataGrid** .  
5 ToString

6  
7 [C#] public const PagerPosition Top;

8 [C++] public: const PagerPosition Top;

9 [VB] Public Const Top As PagerPosition

10 [JScript] public var Top : PagerPosition;

11  
12 *Description*

13 A pager positioned at the top of the  
14 **System.Web.UI.WebControls.DataGrid** .  
15 ToString

16  
17 [C#] public const PagerPosition TopAndBottom;

18 [C++] public: const PagerPosition TopAndBottom;

19 [VB] Public Const TopAndBottom As PagerPosition

20 [JScript] public var TopAndBottom : PagerPosition;

21  
22 *Description*

23 Pagers positioned at both the top and the bottom of the  
24 **System.Web.UI.WebControls.DataGrid** .  
25 Panel class (System.Web.UI.WebControls)

1 ToString

2

3

4 *Description*

5 Represents a control that acts as a container for other controls.

6 The **System.Web.UI.WebControls.Panel** control is a container for other  
7 controls. It is especially useful when you want to generate controls  
8 programmatically or hide/show a group of controls.

9 Panel

10 *Example Syntax:*

11 ToString

12

13 [C#] public Panel();

14 [C++] public: Panel();

15 [VB] Public Sub New()

16 [JScript] public function Panel();

17

18 *Description*

19 Initializes a new instance of the **System.Web.UI.WebControls.Panel**  
20 class.

21 AccessKey

22 Attributes

23 BackColor

24 BackImageUrl

25 ToString

1  
2  
3 *Description*

4 Gets or sets the URL of the background image for the panel control.

5 Use this property to display a custom image for the

6 **System.Web.UI.WebControls.Panel** control.

7 BorderColor

8 BorderStyle

9 BorderWidth

10 ChildControlsCreated

11 ClientID

12 Context

13 Controls

14 ControlStyle

15 ControlStyleCreated

16 CssClass

17 Enabled

18 EnableViewState

19 Events

20 Font

21 ForeColor

22 HasChildViewState

23 Height

24 HorizontalAlign

25 ToString

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

*Description*

Gets or sets the horizontal alignment of the contents within the panel.

Use this property to specify the alignment of the elements in the

**System.Web.UI.WebControls.Panel .**

- ID
- IsTrackingViewState
- NamingContainer
- Page
- Parent
- Site
- Style
- TabIndex
- TagKey
- TagName
- TemplateSourceDirectory
- ToolTip
- UniqueID
- ViewState
- ViewStateIgnoresCase
- Visible
- Width
- Wrap
- ToString

1  
2  
3 *Description*

4 Gets or sets a value indicating whether the content wraps within the panel.

5 Use this property to wrap the contents of the

6 **System.Web.UI.WebControls.Panel .**

7 **AddAttributesToRender**

8  
9 [C#] protected override void AddAttributesToRender(HtmlTextWriter writer);

10 [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);

11 [VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As

12 HtmlTextWriter)

13 [JScript] protected override function AddAttributesToRender(writer :

14 HtmlTextWriter);

15  
16 *Description*

17 Add background-image to list of style attributes to render. Add align and

18 nowrap to list of attributes to render.

19 Placeholder class (System.Web.UI.WebControls)

20 TrackViewState

21  
22  
23 *Description*

24 A container to store dynamically added server controls on the Web page.



1 Use the **System.Web.UI.WebControls.PlaceHolder** control as a container  
2 to store dynamically added server controls to the Web page. The  
3 **System.Web.UI.WebControls.PlaceHolder** control does not produce any visible  
4 output and is only used as a container for other controls on the Web page. You can  
5 use the **System.Web.UI.Control.Controls** collection to add, insert, or remove a  
6 control from the **System.Web.UI.WebControls.PlaceHolder** control.

7 Placeholder

8 *Example Syntax:*

9 TrackViewState

10  
11 [C#] public Placeholder();

12 [C++] public: Placeholder();

13 [VB] Public Sub New()

14 [JScript] public function Placeholder();

15 ChildControlsCreated

16 ClientID

17 Context

18 Controls

19 EnableViewState

20 Events

21 HasChildViewState

22 ID

23 IsTrackingViewState

24 NamingContainer

25 Page

1 Parent  
2 Site  
3 TemplateSourceDirectory  
4 UniqueID  
5 ViewState  
6 ViewStateIgnoresCase  
7 Visible  
8 PlaceholderControlBuilder class (System.Web.UI.WebControls)  
9 TrackViewState  
10 **System.Web.UI.WebControls.PlaceHolder**  
11 PlaceholderControlBuilder  
12 *Example Syntax:*  
13 TrackViewState  
14 ControlType  
15 FChildrenAsProperties  
16 FIsNonParserAccessor  
17 HasAspCode  
18 ID  
19 InDesigner  
20 NamingContainerType  
21 Parser  
22 TagName  
23 AllowWhitespaceLiterals

24  
25 [C#] public override bool AllowWhitespaceLiterals();

1 [C++] public: bool AllowWhitespaceLiterals();

2 [VB] Overrides Public Function AllowWhitespaceLiterals() As Boolean

3 [JScript] public override function AllowWhitespaceLiterals() : Boolean; Specifies

4 whether white space literals are allowed.

5       RadioButton class (System.Web.UI.WebControls)

6       ToString

7  
8  
9 *Description*

10       Represents a radio button control.

11       The **System.Web.UI.WebControls.RadioButton** server control permits  
12 you to intersperse the radio buttons in a group with other content in the page. The  
13 buttons are grouped logically because they all share the same

14 **System.Web.UI.WebControls.RadioButton.GroupName** .

15       RadioButton

16       *Example Syntax:*

17       ToString

18  
19 [C#] public RadioButton();

20 [C++] public: RadioButton();

21 [VB] Public Sub New()

22 [JScript] public function RadioButton();

23  
24 *Description*

1        Initializes a new instance of the  
2   **System.Web.UI.WebControls.RadioButton** class.

3        AccessKey

4        Attributes

5        AutoPostBack

6        BackColor

7        BorderColor

8        BorderStyle

9        BorderWidth

10       Checked

11       ChildControlsCreated

12       ClientID

13       Context

14       Controls

15       ControlStyle

16       ControlStyleCreated

17       CssClass

18       Enabled

19       EnableViewState

20       Events

21       Font

22       ForeColor

23       GroupName

24       ToString

25

1  
2  
3 *Description*

4 Gets or sets the name of the group that the radio button belongs to.

5 Use this property to specify a grouping of radio buttons to create a mutually  
6 exclusive set of controls. The property can be used when only one selection is  
7 possible from a list of available options, as in the case of gender.

8 HasChildViewState

9 Height

10 ID

11 IsTrackingViewState

12 NamingContainer

13 Page

14 Parent

15 Site

16 Style

17 TabIndex

18 TagKey

19 TagName

20 TemplateSourceDirectory

21 Text

22 TextAlign

23 ToolTip

24 UniqueID

25 ViewState

1 ViewStateIgnoresCase  
2 Visible  
3 Width  
4 OnPreRender

5  
6 [C#] protected override void OnPreRender(EventArgs e);  
7 [C++] protected: void OnPreRender(EventArgs\* e);  
8 [VB] Overrides Protected Sub OnPreRender(ByVal e As EventArgs)  
9 [JScript] protected override function OnPreRender(e : EventArgs);  
10

11 *Description*

12 This method is invoked just prior to rendering. Register client script for  
13 handling postback if onChangeHandler is set.

14 IPostBackDataHandler.LoadPostData

15  
16 [C#] bool IPostBackDataHandler.LoadPostData(string postDataKey,  
17 NameValueCollection postCollection);  
18 [C++] bool IPostBackDataHandler::LoadPostData(String\* postDataKey,  
19 NameValueCollection\* postCollection);  
20 [VB] Function LoadPostData(ByVal postDataKey As String, ByVal  
21 postCollection As NameValueCollection) As Boolean Implements  
22 IPostBackDataHandler.LoadPostData  
23 [JScript] function IPostBackDataHandler.LoadPostData(postDataKey : String,  
24 postCollection : NameValueCollection) : Boolean;

25 IPostBackDataHandler.RaisePostDataChangedEvent

```

1
2 [C#] void IPostBackDataHandler.RaisePostDataChangedEvent();
3 [C++] void IPostBackDataHandler::RaisePostDataChangedEvent();
4 [VB] Sub RaisePostDataChangedEvent() Implements
5 IPostBackDataHandler.RaisePostDataChangedEvent
6 [JScript] function IPostBackDataHandler.RaisePostDataChangedEvent();
7     RadioButtonList class (System.Web.UI.WebControls)
8     TrackViewState
9
10

```

### *Description*

Represents a list control that encapsulates a group of radio button controls.

The **System.Web.UI.WebControls.RadioButtonList** control provides page developers with a single-selection radio button group that can be dynamically generated via data binding. It contains an **System.Web.UI.WebControls.ListControl.Items** collection with members that correspond to individual items on the list. To determine which item is selected, test the **System.Web.UI.WebControls.ListControl.SelectedItem** property of the list.

**RadioButtonList**

### *Example Syntax:*

**TrackViewState**

```

23
24 [C#] public RadioButtonList();
25 [C++] public: RadioButtonList();

```

1 [VB] Public Sub New()

2 [JScript] public function RadioButtonList();

3  
4 *Description*

5       Initializes a new instance of a  
6 **System.Web.UI.WebControls.RadioButtonList** class.

7       Use this constructor to create and initialize a new instance of the  
8 **System.Web.UI.WebControls.RadioButtonList** class.

9       AccessKey

10      Attributes

11      AutoPostBack

12      BackColor

13      BorderColor

14      BorderStyle

15      BorderWidth

16      CellPadding

17      TrackViewState

18  
19  
20 *Description*

21       Gets or sets the distance (in pixels) between the border and the contents of  
22 the table cell.

23       This property only applies when the  
24 **System.Web.UI.WebControls.RadioButtonList.RepeatLayout** property is set  
25 to **RepeatLayout.Table** .



1       CellSpacing  
2       TrackViewState  
3  
4   [C#] public virtual int CellSpacing {get; set;}  
5   [C++] public: \_\_property virtual int get\_CellSpacing();public: \_\_property virtual  
6   void set\_CellSpacing(int);  
7   [VB] Overridable Public Property CellSpacing As Integer  
8   [JScript] public function get CellSpacing() : int;public function set  
9   CellSpacing(int);

11   *Description*

12       Gets or sets the distance (in pixels) between adjacent table cells.  
13       This property only applies when the  
14   **System.Web.UI.WebControls.RadioButtonList.RepeatLayout** property is set  
15   to **RepeatLayout.Table** .

16       ChildControlsCreated  
17       ClientID  
18       Context  
19       Controls  
20       ControlStyle  
21       ControlStyleCreated  
22       CssClass  
23       DataMember  
24       DataSource  
25       DataTextField

1	DataTextFormatString
2	DataValueField
3	Enabled
4	EnableViewState
5	Events
6	Font
7	ForeColor
8	HasChildViewState
9	Height
10	ID
11	IsTrackingViewState
12	Items
13	NamingContainer
14	Page
15	Parent
16	RepeatColumns
17	TrackViewState

20 *Description*

21 Gets or sets the number of columns to display in the

22 **System.Web.UI.WebControls.RadioButtonList** control.

23 Use this property to specify the number of columns that display items in the

24 **System.Web.UI.WebControls.RadioButtonList** control. If this property is not

25

set, the **System.Web.UI.WebControls.RadioButtonList** control will display all items in the list in a single column.

RepeatDirection

TrackViewState

[C#] public virtual RepeatDirection RepeatDirection {get; set;}

[C++] public: \_\_property virtual RepeatDirection get\_RepeatDirection();public:

\_\_property virtual void set\_RepeatDirection(RepeatDirection);

[VB] Overridable Public Property RepeatDirection As RepeatDirection

[JScript] public function get RepeatDirection() : RepeatDirection;public function

set RepeatDirection(RepeatDirection);

### *Description*

Gets or sets the direction that the radio buttons within the group are displayed.

Use this property to specify the display direction of the **System.Web.UI.WebControls.RadioButtonList** control.

RepeatLayout

TrackViewState

[C#] public virtual RepeatLayout RepeatLayout {get; set;}

[C++] public: \_\_property virtual RepeatLayout get\_RepeatLayout();public:

\_\_property virtual void set\_RepeatLayout(RepeatLayout);

[VB] Overridable Public Property RepeatLayout As RepeatLayout

[JScript] public function get RepeatLayout() : RepeatLayout;public function set

RepeatLayout(RepeatLayout);

*Description*

Gets or sets the layout of radio buttons within the group.

Use this property to specify whether the items in the

**System.Web.UI.WebControls.RadioButtonList** control are displayed in a table.

If this property is set to **RepeatLayout.Table** , the items in the list are displayed in a table. If this property is set to **RepeatLayout.Flow** , the items in the list are displayed without a table structure.

SelectedIndex

SelectedItem

Site

Style

TabIndex

TagKey

TagName

TemplateSourceDirectory

TextAlign

TrackViewState

*Description*

Gets or sets the text alignment for the radio buttons within the group.

Use this property to specify whether the text associated with the radio

buttons appears on the left or right. If this property is set to **TextAlign.Right** , the

1 text is displayed to the right of the radio button. If this property is set to  
2 **TextAlign.Left** , the text is displayed to the left of the radio button.

3 ToolTip

4 UniqueID

5 ViewState

6 ViewStateIgnoresCase

7 Visible

8 Width

9 CreateControlStyle

10  
11 [C#] protected override Style CreateControlStyle();

12 [C++] protected: Style\* CreateControlStyle();

13 [VB] Overrides Protected Function CreateControlStyle() As Style

14 [JScript] protected override function CreateControlStyle() : Style;

15  
16 *Description*

17 Render

18  
19 [C#] protected override void Render(HtmlTextWriter writer);

20 [C++] protected: void Render(HtmlTextWriter\* writer);

21 [VB] Overrides Protected Sub Render(ByVal writer As HtmlTextWriter)

22 [JScript] protected override function Render(writer : HtmlTextWriter);

23  
24 *Description*

25 IPostBackDataHandler.LoadPostData

```

1
2 [C#] bool IPostBackDataHandler.LoadPostData(string postDataKey,
3 NameValueCollection postCollection);
4 [C++] bool IPostBackDataHandler::LoadPostData(String* postDataKey,
5 NameValueCollection* postCollection);
6 [VB] Function LoadPostData(ByVal postDataKey As String, ByVal
7 postCollection As NameValueCollection) As Boolean Implements
8 IPostBackDataHandler.LoadPostData
9 [JScript] function IPostBackDataHandler.LoadPostData(postDataKey : String,
10 postCollection : NameValueCollection) : Boolean;
11     IPostBackDataHandler.RaisePostDataChangedEvent
12
13 [C#] void IPostBackDataHandler.RaisePostDataChangedEvent();
14 [C++] void IPostBackDataHandler::RaisePostDataChangedEvent();
15 [VB] Sub RaisePostDataChangedEvent() Implements
16 IPostBackDataHandler.RaisePostDataChangedEvent
17 [JScript] function IPostBackDataHandler.RaisePostDataChangedEvent();
18     IRepeatInfoUser.GetItemStyle
19
20 [C#] Style IRepeatInfoUser.GetItemStyle(ListItemType itemType, int
21 repeatIndex);
22 [C++] Style* IRepeatInfoUser::GetItemStyle(ListItemType itemType, int
23 repeatIndex);
24 [VB] Function GetItemStyle(ByVal itemType As ListItemType, ByVal
25 repeatIndex As Integer) As Style Implements IRepeatInfoUser.GetItemStyle

```

```

1 [JScript] function IRepeatInfoUser.GetItemStyle(itemType : ListItemType,
2   repeatIndex : int) : Style;
3     IRepeatInfoUser.RenderItem
4
5 [C#] void IRepeatInfoUser.RenderItem(ListItemType itemType, int repeatIndex,
6   RepeatInfo repeatInfo, HtmlTextWriter writer);
7 [C++] void IRepeatInfoUser::RenderItem(ListItemType itemType, int
8   repeatIndex, RepeatInfo* repeatInfo, HtmlTextWriter* writer);
9 [VB] Sub RenderItem(ByVal itemType As ListItemType, ByVal repeatIndex As
10   Integer, ByVal repeatInfo As RepeatInfo, ByVal writer As HtmlTextWriter)
11   Implements IRepeatInfoUser.RenderItem
12 [JScript] function IRepeatInfoUser.RenderItem(itemType : ListItemType,
13   repeatIndex : int, repeatInfo : RepeatInfo, writer : HtmlTextWriter);
14     RangeValidator class (System.Web.UI.WebControls)
15     TrackViewState
16
17

```

### *Description*

Checks whether the value of an input control is within a specified range of values.

The **System.Web.UI.WebControls.RangeValidator** control tests whether the value of an input control is within a specified range.

RangeValidator

*Example Syntax:*

TrackViewState

1	
2	[C#] public RangeValidator();
3	[C++] public: RangeValidator();
4	[VB] Public Sub New()
5	[JScript] public function RangeValidator();
6	AccessKey
7	Attributes
8	BackColor
9	BorderColor
10	BorderStyle
11	BorderWidth
12	ChildControlsCreated
13	ClientID
14	Context
15	Controls
16	ControlStyle
17	ControlStyleCreated
18	ControlToValidate
19	CssClass
20	Display
21	EnableClientScript
22	Enabled
23	EnableViewState
24	ErrorMessage
25	Events

2003-09-09 10:00:00



1 Font  
 2 ForeColor  
 3 HasChildViewState  
 4 Height  
 5 ID  
 6 IsTrackingViewState  
 7 IsValid  
 8 MaximumValue  
 9 TrackViewState

10  
 11  
 12 *Description*

13 Gets or sets the maximum value of the validation range.

14 Use the **System.Web.UI.WebControls.RangeValidator.MaximumValue**  
 15 property to specify the maximum value of the validation range. If the value  
 16 specified by this property fails to convert to the data type specified by the  
 17 **System.Web.UI.WebControls.BaseCompareValidator.Type** property, an  
 18 exception is thrown.

19 MinimumValue

20 TrackViewState

21  
 22 [C#] public string MinimumValue {get; set;}

23 [C++] public: \_\_property String\* get\_MinimumValue();public: \_\_property void  
 24 set\_MinimumValue(String\*);

25 [VB] Public Property MinimumValue As String

1 [JScript] public function get MinimumValue() : String;public function set  
2 MinimumValue(String);

3  
4 *Description*

5 Gets or sets the minimum value of the validation range.

6 Use the **System.Web.UI.WebControls.RangeValidator.MinimumValue**  
7 property to specify the minimum value of the validation range. If the value  
8 specified by this property fails to convert to the data type specified by the  
9 **System.Web.UI.WebControls.BaseCompareValidator.Type** property, a trace  
10 debug message is generated, the  
11 **System.Web.UI.WebControls.BaseValidator.IsValid** property is set to **true** ,  
12 and no additional processing is performed.

13 NamingContainer

14 Page

15 Parent

16 PropertiesValid

17 RenderUplevel

18 Site

19 Style

20 TabIndex

21 TagKey

22 TagName

23 TemplateSourceDirectory

24 Text

25 ToolTip

1 Type  
 2 UniqueID  
 3 ViewState  
 4 ViewStateIgnoresCase  
 5 Visible  
 6 Width  
 7 AddAttributesToRender

8  
 9 [C#] protected override void AddAttributesToRender(HtmlTextWriter writer);  
 10 [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);  
 11 [VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As  
 12 HtmlTextWriter)  
 13 [JScript] protected override function AddAttributesToRender(writer :  
 14 HtmlTextWriter);

15  
 16 *Description*

17 AddAttributesToRender method AddAttributesToRender method  
 18 ControlPropertiesValid

19  
 20 [C#] protected override bool ControlPropertiesValid();  
 21 [C++] protected: bool ControlPropertiesValid();  
 22 [VB] Overrides Protected Function ControlPropertiesValid() As Boolean  
 23 [JScript] protected override function ControlPropertiesValid() : Boolean;

24  
 25 *Description*

1 This is a check of properties to determine any errors made by the developer  
2 This is a check of properties to determine any errors made by the developer

3 EvaluateIsValid

4  
5 [C#] protected override bool EvaluateIsValid();

6 [C++] protected: bool EvaluateIsValid();

7 [VB] Overrides Protected Function EvaluateIsValid() As Boolean

8 [JScript] protected override function EvaluateIsValid() : Boolean;

9  
10 *Description*

11 EvaluateIsValid method EvaluateIsValid method

12 RegularExpressionValidator class (System.Web.UI.WebControls)

13 Validate

14  
15  
16 *Description*

17 Validates whether the value of an associated input control matches the  
18 pattern specified by a regular expression.

19 The **System.Web.UI.WebControls.RegularExpressionValidator** control  
20 checks whether the value of an input control matches a pattern defined by a  
21 regular expression. This type of validation allows you to check for predictable  
22 sequences of characters, such as those in social security numbers, e-mail  
23 addresses, telephone numbers, and postal codes.

24 RegularExpressionValidator

25 *Example Syntax:*

1	Validate
2	
3	[C#] public RegularExpressionValidator();
4	[C++] public: RegularExpressionValidator();
5	[VB] Public Sub New()
6	[JScript] public function RegularExpressionValidator();
7	AccessKey
8	Attributes
9	BackColor
10	BorderColor
11	BorderStyle
12	BorderWidth
13	ChildControlsCreated
14	ClientID
15	Context
16	Controls
17	ControlStyle
18	ControlStyleCreated
19	ControlToValidate
20	CssClass
21	Display
22	EnableClientScript
23	Enabled
24	EnableViewState
25	ErrorMessage

2008-06-06 14:00:00

1	Events
2	Font
3	ForeColor
4	HasChildViewState
5	Height
6	ID
7	IsTrackingViewState
8	IsValid
9	NamingContainer
10	Page
11	Parent
12	PropertiesValid
13	RenderUplevel
14	Site
15	Style
16	TabIndex
17	TagKey
18	TagName
19	TemplateSourceDirectory
20	Text
21	ToolTip
22	UniqueID
23	ValidationExpression
24	Validate
25	

Page 20 of 20

1  
2  
3 *Description*

4 Gets or sets the regular expression that determines the pattern used to  
5 validate a field.

6 Use this property to specify the pattern used to check for predictable  
7 sequences of characters, such as those in social security numbers, e-mail  
8 addresses, telephone numbers, and postal codes.

9 ViewState

10 ViewStateIgnoresCase

11 Visible

12 Width

13 AddAttributesToRender

14  
15 [C#] protected override void AddAttributesToRender(HtmlTextWriter writer);

16 [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);

17 [VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As  
18 HtmlTextWriter)

19 [JScript] protected override function AddAttributesToRender(writer :  
20 HtmlTextWriter);

21  
22 *Description*

23 AddAttributesToRender method AddAttributesToRender method

24 EvaluateIsValid

1  
2 [C#] protected override bool EvaluateIsValid();

3 [C++] protected: bool EvaluateIsValid();

4 [VB] Overrides Protected Function EvaluateIsValid() As Boolean

5 [JScript] protected override function EvaluateIsValid() : Boolean;

6  
7 *Description*

8 EvaluateIsValid method EvaluateIsValid method

9 RepeatDirection enumeration (System.Web.UI.WebControls)

10 Validate

11  
12  
13 *Description*

14 Specifies the direction in which items of a list control are displayed.

15 The **System.Web.UI.WebControls.RepeatDirection** enumeration  
16 represents the different directions in which the items of a list control can be  
17 displayed.

18 Validate

19  
20 [C#] public const RepeatDirection Horizontal;

21 [C++] public: const RepeatDirection Horizontal;

22 [VB] Public Const Horizontal As RepeatDirection

23 [JScript] public var Horizontal : RepeatDirection;

24  
25 *Description*



Items of a list are displayed horizontally in rows from left to right, then top to bottom, until all items are rendered.

Validate

[C#] public const RepeatDirection Vertical;

[C++] public: const RepeatDirection Vertical;

[VB] Public Const Vertical As RepeatDirection

[JScript] public var Vertical : RepeatDirection;

#### *Description*

Items of a list are displayed vertically in columns from top to bottom, and then left to right, until all items are rendered.

Repeater class (System.Web.UI.WebControls)

ToString

#### *Description*

A databound list control that allows custom layout by repeating a specified template for each item displayed in the list.

The **System.Web.UI.WebControls.Repeater** is a basic templated databound list. It has no built-in layout or styles, so you must explicitly declare all HTML layout, formatting, and style tags within the control's templates.

Repeater

*Example Syntax:*

ToString

```

1
2 [C#] public Repeater();
3 [C++] public: Repeater();
4 [VB] Public Sub New()
5 [JScript] public function Repeater();
6

```

### *Description*

Initializes a new instance of the **System.Web.UI.WebControls.Repeater** class.

AlternatingItemTemplate

ToString

```

13 [C#] public virtual ITemplate AlternatingItemTemplate {get; set;}
14 [C++] public: __property virtual ITemplate*
15 get_AlternatingItemTemplate();public: __property virtual void
16 set_AlternatingItemTemplate(ITemplate*);
17 [VB] Overridable Public Property AlternatingItemTemplate As ITemplate
18 [JScript] public function get AlternatingItemTemplate() : ITemplate;public
19 function set AlternatingItemTemplate(ITemplate);
20

```

### *Description*

Gets or sets the **System.Web.UI.ITemplate** that defines how alternating (zero-based odd-indexed) items are rendered.

1 Use this property to provide a different appearance from the  
2 **System.Web.UI.WebControls.Repeater.ItemTemplate** for alternating items in  
3 the **System.Web.UI.WebControls.Repeater** .

4 ChildControlsCreated

5 ClientID

6 Context

7 Controls

8 ToString

9 DataMember

10 ToString

11  
12  
13 *Description*

14 Gets or sets the specific table in the  
15 **System.Web.UI.WebControls.Repeater.DataSource** to bind to the control.

16 If the **System.Web.UI.WebControls.Repeater.DataSource** contains  
17 multiple tables, use this property to specify the table to bind to the  
18 **System.Web.UI.WebControls.Repeater** .

19 DataSource

20 ToString

21  
22 [C#] public virtual object DataSource {get; set;}

23 [C++] public: \_\_property virtual Object\* get\_DataSource();public: \_\_property  
24 virtual void set\_DataSource(Object\*);

25 [VB] Overridable Public Property DataSource As Object

[JScript] public function get DataSource() : Object;public function set DataSource(Object);

#### *Description*

Gets or sets the data source that provides data for populating the list.

Use this property to specify the source of data to populate the

**System.Web.UI.WebControls.Repeater** control. The

**System.Web.UI.WebControls.Repeater.DataSource** can be any

**System.Collections.IEnumerable** derived object such as a

**System.Data.DataView** for accessing databases, an

**System.Collections.ArrayList** , a **System.Collections.Hashtable** , or an array.

EnableViewState

Events

FooterTemplate

ToString

#### *Description*

Gets or sets the **System.Web.UI.ITemplate** that defines how the control footer is rendered.

Use this property to create a template that controls how the footer of a

**System.Web.UI.WebControls.Repeater** control is displayed.

HasChildViewState

HeaderTemplate

ToString

1  
2  
3 *Description*

4 Gets or sets the **System.Web.UI.ITemplate** that defines how the control  
5 header is rendered.

6 Use this property to create a template that controls how the header of a  
7 **System.Web.UI.WebControls.Repeater** control is displayed.

8 ID

9 IsTrackingViewState

10 Items

11 ToString

12  
13  
14 *Description*

15 Gets a collection of **System.Web.UI.WebControls.RepeaterItem** objects  
16 in the **System.Web.UI.WebControls.Repeater** .

17 The **System.Web.UI.WebControls.Repeater** populates the  
18 **System.Web.UI.WebControls.Repeater.Items** collection by enumerating every  
19 object in its **System.Web.UI.WebControls.Repeater.DataSource** . The  
20 **System.Web.UI.WebControls.Repeater.Items** collection is then used by the  
21 **System.Web.UI.WebControls.Repeater** to render each item in the control.

22 ItemTemplate

23 ToString

24  
25 [C#] public virtual ITemplate ItemTemplate {get; set;}

```

1 [C++] public: __property virtual ITemplate* get_ItemTemplate();public:
2 __property virtual void set_ItemTemplate(ITemplate*);
3 [VB] Overridable Public Property ItemTemplate As ITemplate
4 [JScript] public function get ItemTemplate() : ITemplate;public function set
5 ItemTemplate(ITemplate);

```

#### *Description*

Gets or sets the **System.Web.UI.ITemplate** that defines how items are rendered.

Use this property to create a template that controls how individual items in the **System.Web.UI.WebControls.Repeater** are displayed.

NamingContainer

Page

Parent

SeparatorTemplate

ToString

#### *Description*

Gets or sets the **System.Web.UI.ITemplate** that defines how separators between items are rendered.

Use this property to create a template that controls how separators between items are displayed.

Site

TemplateSourceDirectory

1 UniqueID  
2 ViewState  
3 ViewStateIgnoresCase  
4 Visible  
5 ToString

6  
7  
8 *Description*

9 Occurs when a button is clicked in the  
10 **System.Web.UI.WebControls.Repeater** control.

11 This event is raised when a button in the  
12 **System.Web.UI.WebControls.Repeater** is clicked.

13 ToString

14  
15 [C#] public event RepeaterItemEventHandler ItemCreated;

16 [C++] public: \_\_event RepeaterItemEventHandler\* ItemCreated;

17 [VB] Public Event ItemCreated As RepeaterItemEventHandler

18  
19 *Description*

20 Occurs when an item is created in the  
21 **System.Web.UI.WebControls.Repeater** control.

22 This event is raised when an item is created in the  
23 **System.Web.UI.WebControls.Repeater** .

24 ToString

25

```

1
2 [C#] public event RepeaterItemEventHandler ItemDataBound;
3 [C++] public: __event RepeaterItemEventHandler* ItemDataBound;
4 [VB] Public Event ItemDataBound As RepeaterItemEventHandler
5

```

#### 6 *Description*

7 Occurs after an item in the **System.Web.UI.WebControls.Repeater** is  
8 databound but before it is rendered on the page.

9 This event is raised when an item in the  
10 **System.Web.UI.WebControls.Repeater** is databound.

#### 11 **CreateChildControls**

```

12
13 [C#] protected override void CreateChildControls();
14 [C++] protected: void CreateChildControls();
15 [VB] Overrides Protected Sub CreateChildControls()
16 [JScript] protected override function CreateChildControls();
17

```

#### 18 *Description*

#### 19 **CreateControlHierarchy**

```

20
21 [C#] protected virtual void CreateControlHierarchy(bool useDataSource);
22 [C++] protected: virtual void CreateControlHierarchy(bool useDataSource);
23 [VB] Overridable Protected Sub CreateControlHierarchy(ByVal useDataSource
24 As Boolean)
25 [JScript] protected function CreateControlHierarchy(useDataSource : Boolean);

```



1  
2 *Description*

3 A protected method. Creates a control hierarchy, with or without the data  
4 source as specified. Indicates whether to use the data source or not.

5 CreateItem

6  
7 [C#] protected virtual RepeaterItem CreateItem(int itemIndex, ListItemType  
8 itemType);

9 [C++] protected: virtual RepeaterItem\* CreateItem(int itemIndex, ListItemType  
10 itemType);

11 [VB] Overridable Protected Function CreateItem(ByVal itemIndex As Integer,  
12 ByVal itemType As ListItemType) As RepeaterItem

13 [JScript] protected function CreateItem(itemIndex : int, itemType : ListItemType)  
14 : RepeaterItem;

15  
16 *Description*

17 A protected method. Creates a  
18 **System.Web.UI.WebControls.RepeaterItem** with the specified item type and  
19 location within the **System.Web.UI.WebControls.Repeater** . The specified  
20 location within the **System.Web.UI.WebControls.Repeater** to place the created  
21 item. A **System.Web.UI.WebControls.ListItemType** that represents the  
22 specified type of the **System.Web.UI.WebControls.Repeater** item to create.

23 DataBind

24  
25 [C#] public override void DataBind();

1 [C++] public: void DataBind();

2 [VB] Overrides Public Sub DataBind()

3 [JScript] public override function DataBind();

4  
5 *Description*

6 InitializeItem

7  
8 [C#] protected virtual void InitializeItem(RepeaterItem item);

9 [C++] protected: virtual void InitializeItem(RepeaterItem\* item);

10 [VB] Overridable Protected Sub InitializeItem(ByVal item As RepeaterItem)

11 [JScript] protected function InitializeItem(item : RepeaterItem);

12  
13 *Description*

14 A protected method. Populates iteratively the specified  
15 **System.Web.UI.WebControls.RepeaterItem** with a sub-hierarchy of child  
16 controls. The control to be initialized from an inline template.

17 OnBubbleEvent

18  
19 [C#] protected override bool OnBubbleEvent(object sender, EventArgs e);

20 [C++] protected: bool OnBubbleEvent(Object\* sender, EventArgs\* e);

21 [VB] Overrides Protected Function OnBubbleEvent(ByVal sender As Object,  
22 ByVal e As EventArgs) As Boolean

23 [JScript] protected override function OnBubbleEvent(sender : Object, e :  
24 EventArgs) : Boolean;

1  
2 *Description*

3       OnDataBinding

4  
5 [C#] protected override void OnDataBinding(EventArgs e);

6 [C++] protected: void OnDataBinding(EventArgs\* e);

7 [VB] Overrides Protected Sub OnDataBinding(ByVal e As EventArgs)

8 [JScript] protected override function OnDataBinding(e : EventArgs);

9  
10 *Description*

11       A protected method. Raises the **DataBinding** event.

12       OnItemCommand

13  
14 [C#] protected virtual void OnItemCommand(RepeaterCommandEventArgs e);

15 [C++] protected: virtual void OnItemCommand(RepeaterCommandEventArgs\* e);

16 [VB] Overridable Protected Sub OnItemCommand(ByVal e As

17 RepeaterCommandEventArgs)

18 [JScript] protected function OnItemCommand(e : RepeaterCommandEventArgs);

19  
20 *Description*

21       Raises the **System.Web.UI.WebControls.Repeater.ItemCommand**  
22 event.

23       Raising an event invokes the event handler through a delegate. For more  
24 information, see . The

**System.Web.UI.WebControls.RepeaterCommandEventArgs** that contains the event data.

### OnItemCreated

[C#] protected virtual void OnItemCreated(RepeaterItemEventArgs e);  
[C++] protected: virtual void OnItemCreated(RepeaterItemEventArgs\* e);  
[VB] Overridable Protected Sub OnItemCreated(ByVal e As RepeaterItemEventArgs)  
[JScript] protected function OnItemCreated(e : RepeaterItemEventArgs);

### *Description*

Raises the **System.Web.UI.WebControls.Repeater.ItemCreated** event. Raising an event invokes the event handler through a delegate. For more information, see . The **System.Web.UI.WebControls.RepeaterItemEventArgs** that contains the event data.

### OnItemDataBound

[C#] protected virtual void OnItemDataBound(RepeaterItemEventArgs e);  
[C++] protected: virtual void OnItemDataBound(RepeaterItemEventArgs\* e);  
[VB] Overridable Protected Sub OnItemDataBound(ByVal e As RepeaterItemEventArgs)  
[JScript] protected function OnItemDataBound(e : RepeaterItemEventArgs);

### *Description*

1       Raises the **System.Web.UI.WebControls.Repeater.ItemDataBound**  
2 event.

3       Raising an event invokes the event handler through a delegate. For more  
4 information, see . The **System.Web.UI.WebControls.RepeaterItemEventArgs**  
5 that contains the event data.

6       RepeaterCommandEventArgs class (System.Web.UI.WebControls)

7       TrackViewState

8  
9  
10      *Description*

11       Provides data for the  
12 **System.Web.UI.WebControls.Repeater.ItemCommand** event of a  
13 **System.Web.UI.WebControls.Repeater** . This class cannot be inherited.

14       For a list of initial property values for an instance of  
15 **System.Web.UI.WebControls.RepeaterCommandEventArgs** , see the  
16 **System.Web.UI.WebControls.RepeaterCommandEventArgs.#ctor**  
17 constructor. For more information about handling events, see .

18       RepeaterCommandEventArgs

19      *Example Syntax:*

20       TrackViewState

21  
22 [C#] public RepeaterCommandEventArgs(RepeaterItem item, object  
23 commandSource, CommandEventArgs originalArgs);  
24 [C++] public: RepeaterCommandEventArgs(RepeaterItem\* item, Object\*  
25 commandSource, CommandEventArgs\* originalArgs);

```

1 [VB] Public Sub New(ByVal item As RepeaterItem, ByVal commandSource As
2 Object, ByVal originalArgs As CommandEventArgs)
3 [JScript] public function RepeaterCommandEventArgs(item : RepeaterItem,
4 commandSource : Object, originalArgs : CommandEventArgs);
5

```

#### *Description*

Initializes a new instance of the **System.Web.UI.WebControls.RepeaterCommandEventArgs** class.

The following table shows initial property values for an instance of **System.Web.UI.WebControls.RepeaterCommandEventArgs**. A **System.Web.UI.WebControls.RepeaterItem** that represents an item in the **System.Web.UI.WebControls.Repeater**. The **System.Web.UI.WebControls.RepeaterCommandEventArgs.Item** property is set to this value. The command source. The **System.Web.UI.WebControls.RepeaterCommandEventArgs.CommandSource** property is set to this value. The original event arguments.

CommandArgument

CommandName

CommandSource

TrackViewState

#### *Description*

Gets the source of the command.

Use this property to determine the source of the command.

Item

TrackViewState

[C#] public RepeaterItem Item {get;}

[C++] public: \_\_property RepeaterItem\* get\_Item();

[VB] Public ReadOnly Property Item As RepeaterItem

[JScript] public function get Item() : RepeaterItem;

### *Description*

Gets the **System.Web.UI.WebControls.RepeaterItem** associated with the event.

Use this property to access the properties of the **System.Web.UI.WebControls.RepeaterItem** associated with this event.

RepeaterCommandEventHandler delegate (System.Web.UI.WebControls)

ToString

### *Description*

Represents the method that will handle the **System.Web.UI.WebControls.Repeater.ItemCommand** event of a **System.Web.UI.WebControls.Repeater**. The source of the event. A **System.Web.UI.WebControls.RepeaterCommandEventArgs** that contains the event data.

When you create a **System.Web.UI.WebControls.RepeaterCommandEventHandler** delegate, you

1 identify the method that will handle the event. To associate the event with your  
2 event handler, add an instance of the delegate to the event. The event handler is  
3 called whenever the event occurs, unless you remove the delegate. For more  
4 information about event handler delegates, see .

5 RepeaterItem class (System.Web.UI.WebControls)

6 ToString

7  
8  
9 *Description*

10 Represents an item in the **System.Web.UI.WebControls.Repeater**  
11 control.

12 A **System.Web.UI.WebControls.RepeaterItem** object represents an item  
13 in the **System.Web.UI.WebControls.Repeater** control, such as the heading  
14 section, footer section, or a data item.

15 RepeaterItem

16 *Example Syntax:*

17 ToString

18  
19 [C#] public RepeaterItem(int itemIndex, ListItemType itemType);

20 [C++] public: RepeaterItem(int itemIndex, ListItemType itemType);

21 [VB] Public Sub New(ByVal itemIndex As Integer, ByVal itemType As  
22 ListItemType)

23 [JScript] public function RepeaterItem(itemIndex : int, itemType : ListItemType);

24  
25 *Description*



1        Initializes a new instance of the  
2        **System.Web.UI.WebControls.RepeaterItem** class.

3        Use this constructor to create and initialize a new instance of the  
4        **System.Web.UI.WebControls.RepeaterItem** class. The index of the item in the  
5        **System.Web.UI.WebControls.Repeater** control from the  
6        **System.Web.UI.WebControls.Repeater.Items** collection of the control. One of  
7        the **System.Web.UI.WebControls.ListItemType** values.

8        ChildControlsCreated

9        ClientID

10       Context

11       Controls

12       DataItem

13       ToString

14  
15  
16       *Description*

17       Gets or sets a data item associated with the  
18       **System.Web.UI.WebControls.RepeaterItem** object in the  
19       **System.Web.UI.WebControls.Repeater** control.

20       Use the **System.Web.UI.WebControls.RepeaterItem.DataItem** property  
21       to specify or determine the properties of a data item associated with the  
22       **System.Web.UI.WebControls.RepeaterItem** object in the  
23       **System.Web.UI.WebControls.Repeater** control.

24       EnableViewState

25       Events

HasChildViewState

ID

IsTrackingViewState

ItemIndex

ToString

*Description*

Gets the index of the item in the **System.Web.UI.WebControls.Repeater** control from the **System.Web.UI.WebControls.Repeater.Items** collection of the control.

Use the **System.Web.UI.WebControls.RepeaterItem.ItemIndex** property to determine the index number of the item in the **System.Web.UI.WebControls.Repeater** control from the **System.Web.UI.WebControls.Repeater.Items** collection of the control.

ItemType

ToString

[C#] public virtual ListItemType ItemType {get;}

[C++] public: \_\_property virtual ListItemType get\_ItemType();

[VB] Overridable Public ReadOnly Property ItemType As ListItemType

[JScript] public function get ItemType() : ListItemType;

*Description*

1 Gets the type of the item in the **System.Web.UI.WebControls.Repeater**  
2 control.

3 Use the **System.Web.UI.WebControls.RepeaterItem.ItemType** property  
4 to determine the type of an item in the **System.Web.UI.WebControls.Repeater**  
5 control. The following table lists the different item types.

6 NamingContainer

7 Page

8 Parent

9 Site

10 TemplateSourceDirectory

11 UniqueID

12 ViewState

13 ViewStateIgnoresCase

14 Visible

15 OnBubbleEvent

16  
17 [C#] protected override bool OnBubbleEvent(object source, EventArgs e);

18 [C++] protected: bool OnBubbleEvent(Object\* source, EventArgs\* e);

19 [VB] Overrides Protected Function OnBubbleEvent(ByVal source As Object,  
20 ByVal e As EventArgs) As Boolean

21 [JScript] protected override function OnBubbleEvent(source : Object, e :  
22 EventArgs) : Boolean;

23  
24 *Description*

25 RepeaterItemCollection class (System.Web.UI.WebControls)

## TrackViewState

### *Description*

Represents a collection of **System.Web.UI.WebControls.RepeaterItem** objects in the **System.Web.UI.WebControls.Repeater** control. This class cannot be inherited.

The **System.Web.UI.WebControls.RepeaterItemCollection** class represents a collection of **System.Web.UI.WebControls.RepeaterItem** objects, which in turn represent the data items in a **System.Web.UI.WebControls.Repeater** control. To programmatically retrieve **System.Web.UI.WebControls.RepeaterItem** objects from a **System.Web.UI.WebControls.Repeater** control, use one of the following methods: Use the indexer to get a single **System.Web.UI.WebControls.RepeaterItem** object from the collection using array notation.

### **RepeaterItemCollection**

#### *Example Syntax:*

### TrackViewState

```
[C#] public RepeaterItemCollection(ArrayList items);
```

```
[C++] public: RepeaterItemCollection(ArrayList* items);
```

```
[VB] Public Sub New(ByVal items As ArrayList)
```

```
[JScript] public function RepeaterItemCollection(items : ArrayList);
```

1  
2 *Description*

3       Initializes a new instance of the  
4 **System.Web.UI.WebControls.RepeaterItemCollection** class.

5       Use this constructor to create and initialize a new instance of the  
6 **System.Web.UI.WebControls.RepeaterItemCollection** class. A  
7 **System.Collections.ArrayList** object that contains the items with which to  
8 initialize the collection.

9       Count

10      TrackViewState

11  
12 [C#] public int Count {get;}

13 [C++] public: \_\_property int get\_Count();

14 [VB] Public ReadOnly Property Count As Integer

15 [JScript] public function get Count() : int;

16  
17 *Description*

18       Gets the number of **System.Web.UI.WebControls.RepeaterItem** objects  
19 in the collection.

20       Use the **System.Web.UI.WebControls.RepeaterItemCollection.Count**  
21 property to determine the number of

22 **System.Web.UI.WebControls.RepeaterItem** objects in the

23 **System.Web.UI.WebControls.RepeaterItemCollection** collection. The

24 **System.Web.UI.WebControls.RepeaterItemCollection.Count** property is  
25

commonly used when iterating through the collection to determine the upper bound of the collection.

IsReadOnly

TrackViewState

[C#] public bool IsReadOnly {get;}

[C++] public: \_\_property bool get\_IsReadOnly();

[VB] Public ReadOnly Property IsReadOnly As Boolean

[JScript] public function get IsReadOnly() : Boolean;

### *Description*

Gets a value that indicates whether the **System.Web.UI.WebControls.RepeaterItem** objects in the **System.Web.UI.WebControls.RepeaterItemCollection** can be modified.

This property always returns **false** to indicate that the **System.Web.UI.WebControls.RepeaterItemCollection** can be written to in all cases.

IsSynchronized

TrackViewState

[C#] public bool IsSynchronized {get;}

[C++] public: \_\_property bool get\_IsSynchronized();

[VB] Public ReadOnly Property IsSynchronized As Boolean

[JScript] public function get IsSynchronized() : Boolean;

1  
2 *Description*

3 Gets a value indicating whether access to the  
4 **System.Web.UI.WebControls.RepeaterItemCollection** is synchronized (thread-  
5 safe).

6 This property is derived from the **System.Collections.ICollection** class and  
7 is overridden to always return **false** .

8 Item

9 TrackViewState

10  
11 [C#] public RepeaterItem this[int index] {get;}

12 [C++] public: \_\_property RepeaterItem\* get\_Item(int index);

13 [VB] Public Default ReadOnly Property Item(ByVal index As Integer) As  
14 RepeaterItem

15 [JScript] returnValue = RepeaterItemCollectionObject.Item(index);  
16

17 *Description*

18 Gets the **System.Web.UI.WebControls.RepeaterItem** object at the  
19 specified index in the collection.

20 Use this indexer to get a **System.Web.UI.WebControls.RepeaterItem**  
21 object from the **System.Web.UI.WebControls.RepeaterItemCollection** at the  
22 specified index, using array notation. The zero-based index of the  
23 **System.Web.UI.WebControls.RepeaterItem** object to retrieve in the collection.

24 SyncRoot

25 TrackViewState

1 [C#] public object SyncRoot {get;}

2 [C++] public: \_\_property Object\* get\_SyncRoot();

3 [VB] Public ReadOnly Property SyncRoot As Object

4 [JScript] public function get SyncRoot() : Object;

5  
6  
7 *Description*

8 Gets the object that can be used to synchronize access to the  
9 **System.Web.UI.WebControls.RepeaterItemCollection** collection.

10 The object returned in this implementation is the  
11 **System.Web.UI.WebControls.RepeaterItemCollection** object itself.

12 *CopyTo*

13  
14 [C#] public void CopyTo(Array array, int index);

15 [C++] public: \_\_sealed void CopyTo(Array\* array, int index);

16 [VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As  
17 Integer)

18 [JScript] public function CopyTo(array : Array, index : int);

19  
20 *Description*

21 Copies all the items from this  
22 **System.Web.UI.WebControls.RepeaterItemCollection** to the specified  
23 **System.Array** object, starting at the specified index in the **System.Array** object.

24 Use this method to copy the contents of the  
25 **System.Web.UI.WebControls.RepeaterItemCollection** into the specified



**System.Array** object, starting at the specified index. A zero-based **System.Array** object that receives the copied items from the **System.Web.UI.WebControls.RepeaterItemCollection**. The first position in the specified **System.Array** object to receive the copied contents.

GetEnumerator

[C#] public IEnumerator GetEnumerator();

[C++] public: \_\_sealed IEnumerator\* GetEnumerator();

[VB] NotOverridable Public Function GetEnumerator() As IEnumerator

[JScript] public function GetEnumerator() : IEnumerator;

### *Description*

Returns a **System.Collections.IEnumerator** interface that contains all **System.Web.UI.WebControls.RepeaterItem** objects in the **System.Web.UI.WebControls.RepeaterItemCollection**.

*Return Value:* A **System.Collections.IEnumerator** interface that contains all **System.Web.UI.WebControls.RepeaterItem** objects in the **System.Web.UI.WebControls.RepeaterItemCollection**.

Use this method to create a **System.Collections.IEnumerator** that can be easily iterated through to get each item in the **System.Web.UI.WebControls.RepeaterItemCollection**.

RepeaterItemEventArgs class (System.Web.UI.WebControls)

ToString

1  
2  
3 *Description*

4 Provides data for the  
5 **System.Web.UI.WebControls.Repeater.ItemCreated** and  
6 **System.Web.UI.WebControls.Repeater.ItemDataBound** events of a  
7 **System.Web.UI.WebControls.Repeater** .

8 For a list of initial property values for an instance of  
9 **System.Web.UI.WebControls.RepeaterItemEventArgs** , see the  
10 **System.Web.UI.WebControls.RepeaterItemEventArgs.#ctor** constructor.

11 **RepeaterItemEventArgs**

12 *Example Syntax:*

13 **ToString**

14  
15 [C#] public RepeaterItemEventArgs(RepeaterItem item);

16 [C++] public: RepeaterItemEventArgs(RepeaterItem\* item);

17 [VB] Public Sub New(ByVal item As RepeaterItem)

18 [JScript] public function RepeaterItemEventArgs(item : RepeaterItem);

19  
20 *Description*

21 Initializes a new instance of the  
22 **System.Web.UI.WebControls.RepeaterItemEventArgs** class.

23 The following table shows initial property values for an instance of  
24 **System.Web.UI.WebControls.RepeaterItemEventArgs** . The  
25 **System.Web.UI.WebControls.RepeaterItem** associated with the event. The

1 **System.Web.UI.WebControls.RepeaterItemEventArgs.Item** property is set to  
2 this value.

3 Item

4 ToString

5  
6 [C#] public RepeaterItem Item {get;}

7 [C++] public: \_\_property RepeaterItem\* get\_Item();

8 [VB] Public ReadOnly Property Item As RepeaterItem

9 [JScript] public function get Item() : RepeaterItem;

10  
11 *Description*

12 Gets the **System.Web.UI.WebControls.RepeaterItem** associated with the  
13 event.

14 Use this property to access the properties of the  
15 **System.Web.UI.WebControls.RepeaterItem** associated with this event.

16 RepeaterItemEventHandler delegate (System.Web.UI.WebControls)

17 ToString

18  
19  
20 *Description*

21 Represents the method that will handle the  
22 **System.Web.UI.WebControls.Repeater.ItemCreated** and  
23 **System.Web.UI.WebControls.Repeater.ItemDataBound** events of a  
24 **System.Web.UI.WebControls.Repeater** . The source of the event. A

1 **System.Web.UI.WebControls.RepeaterItemEventArgs** that contains the event  
2 data.

3       When you create a  
4 **System.Web.UI.WebControls.RepeaterItemEventHandler** delegate, you  
5 identify the method that will handle the event. To associate the event with your  
6 event handler, add an instance of the delegate to the event. The event handler is  
7 called whenever the event occurs, unless you remove the delegate. For more  
8 information about event handler delegates, see .

9       RepeatInfo class (System.Web.UI.WebControls)

10       ToString

11  
12  
13 *Description*

14       Defines the information used to render a list of items using a  
15 **System.Web.UI.WebControls.Repeater** .

16       RepeatInfo

17       *Example Syntax:*

18       ToString

19  
20 [C#] public RepeatInfo();

21 [C++] public: RepeatInfo();

22 [VB] Public Sub New()

23 [JScript] public function RepeatInfo();

24  
25 *Description*

1        Initializes a new instance of the  
2        **System.Web.UI.WebControls.RepeatInfo** class. This class is not inheritable.

3        OuterTableImplied

4        ToString

6        [C#] public bool OuterTableImplied {get; set;}

7        [C++] public: \_\_property bool get\_OuterTableImplied();public: \_\_property void  
8        set\_OuterTableImplied(bool);

9        [VB] Public Property OuterTableImplied As Boolean

10        [JScript] public function get OuterTableImplied() : Boolean;public function set  
11        OuterTableImplied(Boolean);

13        *Description*

14        Indicates whether an outer table is implied for the items.

15        RepeatColumns

16        ToString

18        [C#] public int RepeatColumns {get; set;}

19        [C++] public: \_\_property int get\_RepeatColumns();public: \_\_property void  
20        set\_RepeatColumns(int);

21        [VB] Public Property RepeatColumns As Integer

22        [JScript] public function get RepeatColumns() : int;public function set  
23        RepeatColumns(int);

25        *Description*

1 Indicates the column count of items.

2 RepeatDirection

3 ToString

4

5 [C#] public RepeatDirection RepeatDirection {get; set;}

6 [C++] public: \_\_property RepeatDirection get\_RepeatDirection();public:

7 \_\_property void set\_RepeatDirection(RepeatDirection);

8 [VB] Public Property RepeatDirection As RepeatDirection

9 [JScript] public function get RepeatDirection() : RepeatDirection;public function

10 set RepeatDirection(RepeatDirection);

11

12 *Description*

13 Indicates the direction of flow of items.

14 RepeatLayout

15 ToString

16

17 [C#] public RepeatLayout RepeatLayout {get; set;}

18 [C++] public: \_\_property RepeatLayout get\_RepeatLayout();public: \_\_property

19 void set\_RepeatLayout(RepeatLayout);

20 [VB] Public Property RepeatLayout As RepeatLayout

21 [JScript] public function get RepeatLayout() : RepeatLayout;public function set

22 RepeatLayout(RepeatLayout);

23

24 *Description*

25 Indicates the layout of items.

## RenderRepeater

```
[C#] public void RenderRepeater(HtmlTextWriter writer, IRepeatInfoUser user,  
Style controlStyle, WebControl baseControl);  
[C++] public: void RenderRepeater(HtmlTextWriter* writer, IRepeatInfoUser*  
user, Style* controlStyle, WebControl* baseControl);  
[VB] Public Sub RenderRepeater(ByVal writer As HtmlTextWriter, ByVal user  
As IRepeatInfoUser, ByVal controlStyle As Style, ByVal baseControl As  
WebControl)  
[JScript] public function RenderRepeater(writer : HtmlTextWriter, user :  
IRepeatInfoUser, controlStyle : Style, baseControl : WebControl);
```

### *Description*

Renders the Repeater with the specified information. The output stream that renders HTML content to the client. The control style to copy. The control to copy base attributes from.

RepeatLayout enumeration (System.Web.UI.WebControls)

ToString

### *Description*

Specifies the layout of items in a list control.

The **System.Web.UI.WebControls.RepeatLayout** enumeration represents the different layout options for a list control.

ToString

```

1 [C#] public const RepeatLayout Flow;
2
3 [C++] public: const RepeatLayout Flow;
4
5 [VB] Public Const Flow As RepeatLayout
6
7 [JScript] public var Flow : RepeatLayout;
8

```

#### *Description*

Items are displayed without a table structure.

ToString

```

11 [C#] public const RepeatLayout Table;
12
13 [C++] public: const RepeatLayout Table;
14
15 [VB] Public Const Table As RepeatLayout
16
17 [JScript] public var Table : RepeatLayout;
18

```

#### *Description*

Items are displayed in a table.

RequiredFieldValidator class (System.Web.UI.WebControls)

ToString

#### *Description*

Makes the associated input control a required field.

Use this control to make an input control a required field. The input control fails validation if its value does not change from the



1 **System.Web.UI.WebControls.RequiredFieldValidator.InitialValue** property  
2 upon losing focus.

3 RequiredFieldValidator

4 *Example Syntax:*

5 ToString

6  
7 [C#] public RequiredFieldValidator();

8 [C++] public: RequiredFieldValidator();

9 [VB] Public Sub New()

10 [JScript] public function RequiredFieldValidator();

11 AccessKey

12 Attributes

13 BackColor

14 BorderColor

15 BorderStyle

16 BorderWidth

17 ChildControlsCreated

18 ClientID

19 Context

20 Controls

21 ControlStyle

22 ControlStyleCreated

23 ControlToValidate

24 CssClass

25 Display

1	EnableClientScript
2	Enabled
3	EnableViewState
4	ErrorMessage
5	Events
6	Font
7	ForeColor
8	HasChildViewState
9	Height
10	ID
11	InitialValue
12	ToString

*Description*

Gets or sets the initial value of the associated input control.

Use this property to specify the initial value of the input control.

18	IsTrackingViewState
19	IsValid
20	NamingContainer
21	Page
22	Parent
23	PropertiesValid
24	RenderUplevel
25	Site

1 Style  
2 TabIndex  
3 TagKey  
4 TagName  
5 TemplateSourceDirectory  
6 Text  
7 ToolTip  
8 UniqueID  
9 ViewState  
10 ViewStateIgnoresCase  
11 Visible  
12 Width  
13 AddAttributesToRender

14  
15 [C#] protected override void AddAttributesToRender(HtmlTextWriter writer);  
16 [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);  
17 [VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As  
18 HtmlTextWriter)  
19 [JScript] protected override function AddAttributesToRender(writer :  
20 HtmlTextWriter);  
21

22 *Description*

23 AddAttributesToRender method AddAttributesToRender method  
24 EvaluateIsValid  
25

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

[C#] protected override bool EvaluateIsValid();  
[C++] protected: bool EvaluateIsValid();  
[VB] Overrides Protected Function EvaluateIsValid() As Boolean  
[JScript] protected override function EvaluateIsValid() : Boolean;

*Description*

EvaluateIsValid method EvaluateIsValid method  
Table.RowControlCollection class (System.Web.UI.WebControls)  
Validate

*Description*

Count  
IsReadOnly  
IsSynchronized  
Item  
Owner  
SyncRoot  
Add

[C#] public override void Add(Control child);  
[C++] public: void Add(Control\* child);  
[VB] Overrides Public Sub Add(ByVal child As Control)

1 [JScript] public override function Add(child : Control);

3 *Description*

4 Adds the specified **System.Web.UI.Control** object to the collection. The  
5 new control is added to the end of the array.

6 **AddAt**

8 [C#] public override void AddAt(int index, Control child);

9 [C++] public: void AddAt(int index, Control\* child);

10 [VB] Overrides Public Sub AddAt(ByVal index As Integer, ByVal child As  
11 Control)

12 [JScript] public override function AddAt(index : int, child : Control);

14 *Description*

15 Adds the specified **System.Web.UI.Control** object to the collection. The  
16 new control is added to the array at the specified index location. The location in  
17 the array to add the child control. The **Control** object to add to the collection.

18 SelectedDatesCollection class (System.Web.UI.WebControls)

19 **ToString**

22 *Description*

23 Encapsulates a collection of **System.DateTime** objects that represent the  
24 selected dates in a **System.Web.UI.WebControls.Calendar** control. This class  
25 cannot be inherited.

Use this class to programmatically manage a collection of **System.DateTime** objects that represent the selected dates in a **System.Web.UI.WebControls.Calendar** control. This class is commonly used to add or remove dates.

SelectedDatesCollection

*Example Syntax:*

ToString

[C#] public SelectedDatesCollection(ArrayList dateList);

[C++] public: SelectedDatesCollection(ArrayList\* dateList);

[VB] Public Sub New(ByVal dateList As ArrayList)

[JScript] public function SelectedDatesCollection(dateList : ArrayList);

#### *Description*

Initializes a new instance of the **System.Web.UI.WebControls.SelectedDatesCollection** class with the specified date list.

Use this constructor to create a new instance of the **System.Web.UI.WebControls.SelectedDatesCollection** class. A **System.Collections.ArrayList** that represents a collection of dates.

Count

ToString

[C#] public int Count {get;}

[C++] public: \_\_property int get\_Count();

1 [VB] Public ReadOnly Property Count As Integer

2 [JScript] public function get Count() : int;

3  
4 *Description*

5 Gets the number of **System.DateTime** objects in the  
6 **System.Web.UI.WebControls.SelectedDatesCollection** .

7 Use this property to determine the number of dates in the  
8 **System.Web.UI.WebControls.SelectedDatesCollection** . The  
9 **System.Web.UI.WebControls.SelectedDatesCollection.Count** property is often  
10 used when iterating through the collection to determine the upper bound.

11 **ReadOnly**

12 **ToString**

13  
14 [C#] public bool ReadOnly {get;}

15 [C++] public: \_\_property bool get\_ReadOnly();

16 [VB] Public ReadOnly Property ReadOnly As Boolean

17 [JScript] public function get ReadOnly() : Boolean;

18  
19 *Description*

20 Gets a value indicating whether the  
21 **System.Web.UI.WebControls.SelectedDatesCollection** is read-only.

22 This property always returns **false** to indicate that the  
23 **System.Web.UI.WebControls.SelectedDatesCollection** can be written to in all  
24 cases.

25 **IsSynchronized**

ToString

[C#] public bool IsSynchronized {get;}

[C++] public: \_\_property bool get\_IsSynchronized();

[VB] Public ReadOnly Property IsSynchronized As Boolean

[JScript] public function get IsSynchronized() : Boolean;

### *Description*

Gets a value indicating whether access to the **System.Web.UI.WebControls.SelectedDatesCollection** is synchronized (thread-safe).

This property is derived from **System.Collections.ICollection** and is overridden to always return **false**.

Item

ToString

[C#] public DateTime this[int index] {get;}

[C++] public: \_\_property DateTime get\_Item(int index);

[VB] Public Default ReadOnly Property Item(ByVal index As Integer) As  
DateTime

[JScript] returnValue = SelectedDatesCollectionObject.Item(index);

### *Description*

Gets a **System.DateTime** at the specified index in the **System.Web.UI.WebControls.SelectedDatesCollection**.



1 Use this indexer to get an individual **System.DateTime** in the  
2 **System.Web.UI.WebControls.SelectedDatesCollection** at the specified index  
3 using simple array notation. An ordinal index value that specifies which  
4 **System.DateTime** to return.

5 SyncRoot

6 ToString

7  
8 [C#] public object SyncRoot {get;}

9 [C++] public: \_\_property Object\* get\_SyncRoot();

10 [VB] Public ReadOnly Property SyncRoot As Object

11 [JScript] public function get SyncRoot() : Object;

12  
13 *Description*

14 Gets the object that can be used to synchronize access to the  
15 **System.Web.UI.WebControls.SelectedDatesCollection** .

16 The object returned in this implementation is the  
17 **System.Web.UI.WebControls.SelectedDatesCollection** object itself.

18 Add

19  
20 [C#] public void Add(DateTime date);

21 [C++] public: void Add(DateTime date);

22 [VB] Public Sub Add(ByVal date As DateTime)

23 [JScript] public function Add(date : DateTime);

24  
25 *Description*

1 Appends the specified **System.DateTime** to the end of the  
2 **System.Web.UI.WebControls.SelectedDatesCollection** .

3 Use this method to add the specified **System.DateTime** to the end of the  
4 **System.Web.UI.WebControls.SelectedDatesCollection** . The **System.DateTime**  
5 to add to the collection.

6 Clear

7  
8 [C#] public void Clear();

9 [C++] public: void Clear();

10 [VB] Public Sub Clear()

11 [JScript] public function Clear();

12  
13 *Description*

14 Removes all **System.DateTime** controls from the collection.

15 Use this method to remove all **System.DateTime** objects from the  
16 **System.Web.UI.WebControls.SelectedDatesCollection** and set the  
17 **System.Web.UI.WebControls.SelectedDatesCollection.Count** property to 0.

18 Contains

19  
20 [C#] public bool Contains(DateTime date);

21 [C++] public: bool Contains(DateTime date);

22 [VB] Public Function Contains(ByVal date As DateTime) As Boolean

23 [JScript] public function Contains(date : DateTime) : Boolean;

24  
25 *Description*

1 Returns a value indicating whether the  
2 **System.Web.UI.WebControls.SelectedDatesCollection** contains the specified  
3 **System.DateTime** object.

4 *Return Value:* **true** if the  
5 **System.Web.UI.WebControls.SelectedDatesCollection** contains the specified  
6 **System.DateTime** ; otherwise, **false** .

7 Use this method to determine whether the  
8 **System.Web.UI.WebControls.SelectedDatesCollection** contains the specified  
9 **System.DateTime** . The **System.DateTime** object to search for in the  
10 **System.Web.UI.WebControls.SelectedDatesCollection**.

11 **CopyTo**

12  
13 [C#] public void CopyTo(Array array, int index);

14 [C++] public: \_\_sealed void CopyTo(Array\* array, int index);

15 [VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As  
16 Integer)

17 [JScript] public function CopyTo(array : Array, index : int);  
18

### 19 *Description*

20 Copies the items from the  
21 **System.Web.UI.WebControls.SelectedDatesCollection** to the specified  
22 **System.Array** , starting with the specified index.

23 Use this method to copy the contents of the  
24 **System.Web.UI.WebControls.SelectedDatesCollection** into the specified  
25 **System.Array** , starting at the specified index. A zero-based **System.Array** that

1 receives the copied items from the

2 **System.Web.UI.WebControls.SelectedDatesCollection**. The first index in the  
3 specified **System.Array** to receive the items.

4 **GetEnumerator**

5  
6 [C#] public IEnumerator GetEnumerator();

7 [C++] public: \_\_sealed IEnumerator\* GetEnumerator();

8 [VB] NotOverridable Public Function GetEnumerator() As IEnumerator

9 [JScript] public function GetEnumerator() : IEnumerator;

10  
11 *Description*

12 Returns a **System.Collections.IEnumerator** that contains all  
13 **System.DateTime** objects within the  
14 **System.Web.UI.WebControls.SelectedDatesCollection** .

15 *Return Value:* A **System.Collections.IEnumerator** that contains all  
16 **System.DateTime** objects within the  
17 **System.Web.UI.WebControls.SelectedDatesCollection** .

18 Use this method to create a **System.Collections.IEnumerator** that can be  
19 easily iterated through to get each item in the  
20 **System.Web.UI.WebControls.SelectedDatesCollection** .

21 **Remove**

22  
23 [C#] public void Remove(DateTime date);

24 [C++] public: void Remove(DateTime date);

25 [VB] Public Sub Remove(ByVal date As DateTime)

1 [JScript] public function Remove(date : DateTime);

2  
3 *Description*

4 Removes the specified **System.DateTime** object from the  
5 **System.Web.UI.WebControls.SelectedDatesCollection** .

6 Use this method to remove the specified **System.DateTime** from the  
7 **System.Web.UI.WebControls.SelectedDatesCollection** . The **System.DateTime**  
8 object to remove from the  
9 **System.Web.UI.WebControls.SelectedDatesCollection**.

10 **SelectRange**

11  
12 [C#] public void SelectRange(DateTime fromDate, DateTime toDate);

13 [C++] public: void SelectRange(DateTime fromDate, DateTime toDate);

14 [VB] Public Sub SelectRange(ByVal fromDate As DateTime, ByVal toDate As  
15 DateTime)

16 [JScript] public function SelectRange(fromDate : DateTime, toDate : DateTime);

17  
18 *Description*

19 Adds the specified range of dates to the  
20 **System.Web.UI.WebControls.SelectedDatesCollection** .

21 Use this method to add the specified range of dates to the  
22 **System.Web.UI.WebControls.SelectedDatesCollection** . A **System.DateTime**  
23 object that specifies the initial date to add to the  
24 **System.Web.UI.WebControls.SelectedDatesCollection**. A **System.DateTime**

object that specifies the end date to add to the

**System.Web.UI.WebControls.SelectedDatesCollection.**

ServerValidateEventArgs class (System.Web.UI.WebControls)

ToString

### *Description*

Provides data for the

**System.Web.UI.WebControls.CustomValidator.ServerValidate** event of the **System.Web.UI.WebControls.CustomValidator** control. This class cannot be inherited.

A **System.Web.UI.WebControls.ServerValidateEventArgs** object is passed to the **System.Web.UI.WebControls.CustomValidator.ServerValidate** event handler to provide event data to the handler. The **System.Web.UI.WebControls.CustomValidator.ServerValidate** event is raised when validation is performed on the server. This allows you to perform a custom server-side validation routine on the value of an input control (with a **System.Web.UI.WebControls.CustomValidator** control associated with it) in the event handler.

ServerValidateEventArgs

*Example Syntax:*

ToString

[C#] public ServerValidateEventArgs(string value, bool isValid);

[C++] public: ServerValidateEventArgs(String\* value, bool isValid);

1 [VB] Public Sub New(ByVal value As String, ByVal isValid As Boolean)  
2 [JScript] public function ServerValidateEventArgs(value : String, isValid :  
3 Boolean);

4  
5 *Description*

6       Initializes a new instance of the  
7 **System.Web.UI.WebControls.ServerValidateEventArgs** class.

8       Use this constructor to create and initialize a new instance of the  
9 **System.Web.UI.WebControls.ServerValidateEventArgs** class. The value to  
10 validate. **true** to indicate that the value passes validation; otherwise, **false**.

11       IsValid

12       ToString

13  
14 [C#] public bool IsValid {get; set;}

15 [C++] public: \_\_property bool get\_IsValid();public: \_\_property void  
16 set\_IsValid(bool);

17 [VB] Public Property IsValid As Boolean

18 [JScript] public function get IsValid() : Boolean;public function set  
19 IsValid(Boolean);

20  
21 *Description*

22       Gets or sets whether the value specified by the  
23 **System.Web.UI.WebControls.ServerValidateEventArgs.Value** property passes  
24 validation.

Once your validation routine finishes, use the **System.Web.UI.WebControls.ServerValidateEventArgs.IsValid** property to indicate whether the value specified by the **System.Web.UI.WebControls.ServerValidateEventArgs.Value** property passes validation. This value determines whether input control associated with the **System.Web.UI.WebControls.CustomValidator** control passes validation.

Value

ToString

[C#] public string Value {get;}

[C++] public: \_\_property String\* get\_Value();

[VB] Public ReadOnly Property Value As String

[JScript] public function get Value() : String;

#### *Description*

Gets the value to validate in the custom event handler for the **System.Web.UI.WebControls.CustomValidator.ServerValidate** event.

Use the **System.Web.UI.WebControls.ServerValidateEventArgs.Value** property determine the value to validate in the custom event handler for the **System.Web.UI.WebControls.CustomValidator.ServerValidate** event. Notice that you cannot programmatically change this value.

ServerValidateEventHandler delegate (System.Web.UI.WebControls)

ToString



1  
2  
3 *Description*

4 Represents the method that will handle the  
5 **System.Web.UI.WebControls.CustomValidator.ServerValidate** event of a  
6 **System.Web.UI.WebControls.CustomValidator** control. The source of the  
7 event. A **System.Web.UI.WebControls.ServerValidateEventArgs** that contains  
8 the event data.

9 Use this delegate to provide custom code that performs validation on the  
10 server. Your code needs to validate the  
11 **System.Web.UI.WebControls.ServerValidateEventArgs.Value** property of the  
12 *args* parameter passed to the delegate. Store the result of the validation in the  
13 **System.Web.UI.WebControls.ServerValidateEventArgs.IsValid** property of  
14 the *args* parameter.

15 Style class (System.Web.UI.WebControls)

16 ToString

17  
18  
19 *Description*

20 Represents attributes that define the style of a Web Forms control and  
21 provides methods to manage the style.

22 Style

23 *Example Syntax:*

24 ToString

1  
2 [C#] public Style();

3 [C++] public: Style();

4 [VB] Public Sub New()

5 [JScript] public function Style(); Initializes a new instance of the  
6 **System.Web.UI.WebControls.Style** class without a specified state bag.

7  
8 *Description*

9       Initializes a new instance of the **System.Web.UI.WebControls.Style** class.

10       Style

11       *Example Syntax:*

12       ToString

13  
14 [C#] public Style(StateBag bag);

15 [C++] public: Style(StateBag\* bag);

16 [VB] Public Sub New(ByVal bag As StateBag)

17 [JScript] public function Style(bag : StateBag);

18  
19 *Description*

20       Initializes a new instance of the **System.Web.UI.WebControls.Style** class  
21 with the specified state bag information. Contains the specified state bag  
22 information.

23       BackColor

24       ToString

1  
2 [C#] public Color BackColor {get; set;}

3 [C++] public: \_\_property Color get\_BackColor();public: \_\_property void

4 set\_BackColor(Color);

5 [VB] Public Property BackColor As Color

6 [JScript] public function get BackColor() : Color;public function set

7 BackColor(Color);

8  
9 *Description*

10 Gets or sets the background color property of the

11 **System.Web.UI.WebControls.Style** class.

12 BorderColor

13 ToString

14  
15 [C#] public Color BorderColor {get; set;}

16 [C++] public: \_\_property Color get\_BorderColor();public: \_\_property void

17 set\_BorderColor(Color);

18 [VB] Public Property BorderColor As Color

19 [JScript] public function get BorderColor() : Color;public function set

20 BorderColor(Color);

21  
22 *Description*

23 Gets or sets the border color property of the

24 **System.Web.UI.WebControls.Style** class.

25 BorderStyle

ToString

[C#] public BorderStyle BorderStyle {get; set;}

[C++] public: \_\_property BorderStyle get\_BorderStyle();public: \_\_property void  
set\_BorderStyle(BorderStyle);

[VB] Public Property BorderStyle As BorderStyle

[JScript] public function get BorderStyle() : BorderStyle;public function set  
BorderStyle(BorderStyle);

### *Description*

Gets or sets the border style property of the  
**System.Web.UI.WebControls.Style** class.

BorderWidth

ToString

[C#] public Unit BorderWidth {get; set;}

[C++] public: \_\_property Unit get\_BorderWidth();public: \_\_property void  
set\_BorderWidth(Unit);

[VB] Public Property BorderWidth As Unit

[JScript] public function get BorderWidth() : Unit;public function set  
BorderWidth(Unit);

### *Description*

Gets or sets the border width property of the  
**System.Web.UI.WebControls.Style** class.

1 Container  
2 CssClass  
3 ToString  
4  
5

6 *Description*

7 Gets or sets the CSS class property of the  
8 **System.Web.UI.WebControls.Style** class.

9 DesignMode  
10 Events  
11 Font  
12 ToString  
13  
14

15 *Description*

16 Gets a **System.Web.UI.WebControls.FontInfo** object that contains the  
17 font properties for the **System.Web.UI.WebControls.Style** class.

18 ForeColor  
19 ToString  
20

21 [C#] public Color ForeColor {get; set;}

22 [C++] public: \_\_property Color get\_ForeColor();public: \_\_property void  
23 set\_ForeColor(Color);

24 [VB] Public Property ForeColor As Color

25 [JScript] public function get ForeColor() : Color;public function set

1 ForeColor(Color);

2  
3 *Description*

4 Gets or sets the foreground color (typically the color of the text) property of  
5 the **System.Web.UI.WebControls.Style** class.

6 Height

7 ToString

8  
9 [C#] public Unit Height {get; set;}

10 [C++] public: \_\_property Unit get\_Height();public: \_\_property void  
11 set\_Height(Unit);

12 [VB] Public Property Height As Unit

13 [JScript] public function get Height() : Unit;public function set Height(Unit);

14  
15 *Description*

16 Gets or sets the height property of the **System.Web.UI.WebControls.Style**  
17 class.

18 IsEmpty

19 ToString

20  
21 [C#] protected internal virtual bool IsEmpty {get;}

22 [C++] internal: \_\_property virtual bool get\_IsEmpty();

23 [VB] Overridable Protected Friend ReadOnly Property IsEmpty As Boolean

24 [JScript] package function get IsEmpty() : Boolean;

1  
2 *Description*

3 A protected property. Gets a value indicating whether any style elements  
4 have been defined in the state bag.

5 IsTrackingViewState

6 ToString

7  
8 [C#] protected bool IsTrackingViewState {get;}

9 [C++] protected: \_\_property bool get\_IsTrackingViewState();

10 [VB] Protected ReadOnly Property IsTrackingViewState As Boolean

11 [JScript] protected function get IsTrackingViewState() : Boolean;

12  
13 *Description*

14 A protected method. Returns a value indicating whether any style elements  
15 have been defined in the state bag.

16 *Return Value:* **true** if there are style elements defined in the state bag; otherwise,  
17 **false** .

18 Site

19 ViewState

20 ToString

21  
22  
23 *Description*

24 Gets the state bag that holds the style elements.

25 The private field name for the state bag is case sensitive.

Width

ToString

[C#] public Unit Width {get; set;}

[C++] public: \_\_property Unit get\_ Width();public: \_\_property void

set\_ Width(Unit);

[VB] Public Property Width As Unit

[JScript] public function get Width() : Unit;public function set Width(Unit);

### *Description*

Gets or sets the width property of the **System.Web.UI.WebControls.Style** class.

AddAttributesToRender

[C#] public void AddAttributesToRender(HtmlTextWriter writer);

[C++] public: void AddAttributesToRender(HtmlTextWriter\* writer);

[VB] Public Sub AddAttributesToRender(ByVal writer As HtmlTextWriter)

[JScript] public function AddAttributesToRender(writer : HtmlTextWriter);

### *Description*

Adds all non-blank style attributes to the HTML output stream to be rendered to the client. The output stream that renders HTML content to the client.

AddAttributesToRender

[C#] public virtual void AddAttributesToRender(HtmlTextWriter writer,



```

1 WebControl owner);
2 [C++] public: virtual void AddAttributesToRender(HtmlTextWriter* writer,
3 WebControl* owner);
4 [VB] Overridable Public Sub AddAttributesToRender(ByVal writer As
5 HtmlTextWriter, ByVal owner As WebControl)
6 [JScript] public function AddAttributesToRender(writer : HtmlTextWriter, owner
7 : WebControl);
8

```

### *Description*

Adds all non-blank style attributes to the HTML output stream to be rendered to the client.

### *CopyFrom*

```

14 [C#] public virtual void CopyFrom(Style s);
15 [C++] public: virtual void CopyFrom(Style* s);
16 [VB] Overridable Public Sub CopyFrom(ByVal s As Style)
17 [JScript] public function CopyFrom(s : Style);
18

```

### *Description*

Copies non-blank elements from the specified style, overwriting existing style elements if necessary. The style to be copied.

### *LoadViewState*

```

24 [C#] protected internal void LoadViewState(object state);
25 [C++] protected public: void LoadViewState(Object* state);

```

1 [VB] Protected Friend Dim Sub LoadViewState(ByVal state As Object)

2 [JScript] package function LoadViewState(state : Object);

3  
4 *Description*

5 A protected method. Load the previously saved state. The previously saved  
6 state.

7 MergeWith

8  
9 [C#] public virtual void MergeWith(Style s);

10 [C++] public: virtual void MergeWith(Style\* s);

11 [VB] Overridable Public Sub MergeWith(ByVal s As Style)

12 [JScript] public function MergeWith(s : Style);

13  
14 *Description*

15 Copies non-blank elements from the specified style, but will not overwrite  
16 any existing style elements. The style to be copied.

17 Reset

18  
19 [C#] public virtual void Reset();

20 [C++] public: virtual void Reset();

21 [VB] Overridable Public Sub Reset()

22 [JScript] public function Reset();

23  
24 *Description*

25 Clears out any defined style elements from the state bag.

## SaveViewState

[C#] protected internal virtual object SaveViewState();  
[C++] protected public: virtual Object\* SaveViewState();  
[VB] Overridable Protected Friend Dim Function SaveViewState() As Object  
[JScript] package function SaveViewState() : Object;

### *Description*

A protected method. Saves any state that has been modified after the **System.Web.UI.WebControls.Style.TrackViewState** method was invoked.

*Return Value:* An object that represents the saved state. The default is **null**.

## SetBit

[C#] protected internal virtual void SetBit(int bit);  
[C++] protected public: virtual void SetBit(int bit);  
[VB] Overridable Protected Friend Dim Sub SetBit(ByVal bit As Integer)  
[JScript] package function SetBit(bit : int);

### *Description*

A protected internal method.

## StateManager.LoadViewState

[C#] void IStateManager.LoadViewState(object state);  
[C++] void IStateManager::LoadViewState(Object\* state);  
[VB] Sub LoadViewState(ByVal state As Object) Implements

```

1 IStateManager.LoadViewState
2 [JScript] function IStateManager.LoadViewState(state : Object);
3     IStateManager.SaveViewState
4
5 [C#] object IStateManager.SaveViewState();
6 [C++] Object* IStateManager::SaveViewState();
7 [VB] Function SaveViewState() As Object Implements
8 IStateManager.SaveViewState
9 [JScript] function IStateManager.SaveViewState() : Object;
10     IStateManager.TrackViewState
11
12 [C#] void IStateManager.TrackViewState();
13 [C++] void IStateManager::TrackViewState();
14 [VB] Sub TrackViewState() Implements IStateManager.TrackViewState
15 [JScript] function IStateManager.TrackViewState();
16     ToString
17
18 [C#] public override string ToString();
19 [C++] public: String* ToString();
20 [VB] Overrides Public Function ToString() As String
21 [JScript] public override function ToString() : String;
22

```

### *Description*

Overrides the **ToString** method to return **System.String.Empty** .

*Return Value:* **System.String.Empty** for all cases.

## TrackViewState

[C#] protected internal virtual void TrackViewState();

[C++] protected public: virtual void TrackViewState();

[VB] Overridable Protected Friend Dim Sub TrackViewState()

[JScript] package function TrackViewState();

### *Description*

A protected method. Marks the beginning for tracking state changes on the control. Any changes made after "mark" will be tracked and saved as part of the control viewstate.

Table class (System.Web.UI.WebControls)

TrackViewState

### *Description*

Constructs a table and defines its properties.

The **System.Web.UI.WebControls.Table** class allows you to build an HTML table and specify its characteristics in a straightforward manner, using the same abstract object model for building any other Web controls. A table can be built at design time given some static contents, but the power of a **System.Web.UI.WebControls.Table** Web control is often realized when the table is built programmatically with dynamic contents. In addition, as for other Web controls, the same code to render a table yields appropriate output accordingly for both downleveling and upleveling browsers.

1       Table

2       *Example Syntax:*

3       TrackViewState

4

5   [C#] public Table();

6   [C++] public: Table();

7   [VB] Public Sub New()

8   [JScript] public function Table();

9

10   *Description*

11       Initializes a new instance of the **System.Web.UI.WebControls.Table**

12   class.

13       AccessKey

14       Attributes

15       BackColor

16       BackImageUrl

17       TrackViewState

18

19

20   *Description*

21       Indicates the URL of the background image to display behind the table. The

22   image will be tiled if it is smaller than the table.

23       BorderColor

24       BorderStyle

25       BorderWidth

1       CellPadding  
2       TrackViewState

3  
4

5       *Description*

6           Gets or sets the distance (in pixels) between the border and the contents of  
7 the table cell.

8       CellSpacing  
9       TrackViewState

10

11      [C#] public virtual int CellSpacing {get; set;}

12      [C++] public: \_\_property virtual int get\_CellSpacing();public: \_\_property virtual  
13 void set\_CellSpacing(int);

14      [VB] Overridable Public Property CellSpacing As Integer

15      [JScript] public function get CellSpacing() : int;public function set  
16 CellSpacing(int);

17

18       *Description*

19           Gets or sets the distance (in pixels) between table cells.

20       ChildControlsCreated

21       ClientID

22       Context

23       Controls

24       ControlStyle

25       ControlStyleCreated

1      CssClass  
2      Enabled  
3      EnableViewState  
4      Events  
5      Font  
6      ForeColor  
7      GridLines  
8      TrackViewState

11      *Description*

12      Gets or sets the gridlines property of the  
13      **System.Web.UI.WebControls.Table** class.

14      HasChildViewState  
15      Height  
16      HorizontalAlign  
17      TrackViewState

20      *Description*

21      Gets or sets the horizontal alignment of the table within the page.  
22      ID  
23      IsTrackingViewState  
24      NamingContainer  
25      Page



1 Parent  
2 Rows  
3 TrackViewState  
4  
5

6 *Description*

7 Gets the collection of rows within the table.

8 This property is only used when building tables programmatically. At  
9 design time, the property is set by declaring **TableRow** Web controls.

10 Site

11 Style

12 TabIndex

13 TagKey

14 TagName

15 TemplateSourceDirectory

16 ToolTip

17 UniqueID

18 ViewState

19 ViewStateIgnoresCase

20 Visible

21 Width

22 AddAttributesToRender  
23

24 [C#] protected override void AddAttributesToRender(HtmlTextWriter writer);

25 [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);

1 [VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As  
2 HtmlTextWriter)

3 [JScript] protected override function AddAttributesToRender(writer :  
4 HtmlTextWriter);

5  
6 *Description*

7 A protected method. Adds information about the border color and border  
8 width HTML attributes to the list of attributes to render. The output stream that  
9 renders HTML content to the client.

10 CreateControlCollection

11  
12 [C#] protected override ControlCollection CreateControlCollection();

13 [C++] protected: ControlCollection\* CreateControlCollection();

14 [VB] Overrides Protected Function CreateControlCollection() As  
15 ControlCollection

16 [JScript] protected override function CreateControlCollection() :  
17 ControlCollection;

18  
19 *Description*

20  
21 CreateControlStyle

22  
23 [C#] protected override Style CreateControlStyle();

24 [C++] protected: Style\* CreateControlStyle();

25 [VB] Overrides Protected Function CreateControlStyle() As Style

1 [JScript] protected override function CreateControlStyle() : Style;

3 *Description*

4 A protected method. Creates a table control style.

5 *Return Value:* A **System.Web.UI.WebControls.Style** that specifies the table  
6 control style.

7 RenderContents

9 [C#] protected override void RenderContents(HtmlTextWriter writer);

10 [C++] protected: void RenderContents(HtmlTextWriter\* writer);

11 [VB] Overrides Protected Sub RenderContents(ByVal writer As HtmlTextWriter)

12 [JScript] protected override function RenderContents(writer : HtmlTextWriter);

14 *Description*

15 A protected method. The output stream that renders HTML content to the  
16 client.

17 TableCell class (System.Web.UI.WebControls)

18 TrackViewState

21 *Description*

22 Represents a cell in a **System.Web.UI.WebControls.Table** control.

23 The **System.Web.UI.WebControls.TableCell** class represents a cell in a  
24 **System.Web.UI.WebControls.Table** control. You can use the

**System.Web.UI.WebControls.TableCell.Text** property to specify or determine the contents of the cell.

TableCell

*Example Syntax:*

TrackViewState

[C#] public TableCell();

[C++] public: TableCell();

[VB] Public Sub New()

[JScript] public function TableCell();

### *Description*

Initializes a new instance of the **System.Web.UI.WebControls.TableCell** class.

Use this constructor to create and initialize a new instance of the **System.Web.UI.WebControls.TableCell** class.

AccessKey

Attributes

BackColor

BorderColor

BorderStyle

BorderWidth

ChildControlsCreated

ClientID

ColumnSpan

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

TrackViewState

*Description*

Gets or sets the number of columns in the **System.Web.UI.WebControls.Table** control that the cell spans.

Use the **System.Web.UI.WebControls.TableCell.ColumnSpan** property to specify or determine the number of columns in the rendered table that the cell spans. For example, if you set this property to **2** , the cell takes up two columns in the **System.Web.UI.WebControls.Table** control.

Context

Controls

ControlStyle

ControlStyleCreated

CssClass

Enabled

EnableViewState

Events

Font

ForeColor

HasChildViewState

Height

HorizontalAlign

TrackViewState

1  
2  
3 *Description*

4 Gets or sets the horizontal alignment of the contents in the cell.

5 Use the **System.Web.UI.WebControls.TableCell.HorizontalAlign**  
6 property to specify the horizontal alignment of the contents of the cell. The  
7 following table lists the possible values.

8 ID

9 IsTrackingViewState

10 NamingContainer

11 Page

12 Parent

13 RowSpan

14 TrackViewState

15  
16  
17 *Description*

18 Gets or sets the number of rows in the

19 **System.Web.UI.WebControls.Table** control that the cell spans.

20 Use the **System.Web.UI.WebControls.TableCell.RowSpan** property to  
21 specify or determine the number of rows in the rendered table that the cell spans.

22 For example, if you set this property to **2** , the cell takes up two rows in the

23 **System.Web.UI.WebControls.Table** control.

24 Site

25 Style

1 TabIndex  
2 TagKey  
3 TagName  
4 TemplateSourceDirectory  
5 Text  
6 TrackViewState

7  
8  
9 *Description*

10 Gets or sets the text contents of the cell.  
11 Use the **System.Web.UI.WebControls.TableCell.Text** property to specify  
12 or determine the text contents of the cell. This property is commonly used to  
13 programmatically update the contents of a cell.

14 ToolTip  
15 UniqueID  
16 VerticalAlign  
17 TrackViewState

18  
19  
20 *Description*

21 Gets or sets the vertical alignment of the contents in the cell.  
22 Use the **System.Web.UI.WebControls.TableCell.VerticalAlign** property  
23 to specify the vertical alignment of the contents of the cell. The following table  
24 lists the possible values.

25 ViewState

1 ViewStateIgnoresCase

2 Visible

3 Width

4 Wrap

5 TrackViewState

6  
7  
8 *Description*

9 Gets or sets a value that indicates whether the content of the cell wrap in the  
10 cell.

11 Use the **System.Web.UI.WebControls.TableCell.Wrap** property to  
12 specify or determine whether the content of the cell wrap in the cell.

13 AddAttributesToRender

14  
15 [C#] protected override void AddAttributesToRender(HtmlTextWriter writer);

16 [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);

17 [VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As  
18 HtmlTextWriter)

19 [JScript] protected override function AddAttributesToRender(writer :  
20 HtmlTextWriter);

21  
22 *Description*

23 A protected method. Adds information about the column span and row span  
24 to the list of attributes to render. The output stream that renders HTML content to  
25 the client.



## AddParsedSubObject

[C#] protected override void AddParsedSubObject(object obj);  
[C++] protected: void AddParsedSubObject(Object\* obj);  
[VB] Overrides Protected Sub AddParsedSubObject(ByVal obj As Object)  
[JScript] protected override function AddParsedSubObject(obj : Object);

## CreateControlStyle

[C#] protected override Style CreateControlStyle();  
[C++] protected: Style\* CreateControlStyle();  
[VB] Overrides Protected Function CreateControlStyle() As Style  
[JScript] protected override function CreateControlStyle() : Style;

### *Description*

A protected method. Creates a table item control style.

*Return Value:* A **System.Web.UI.WebControls.Style** that specifies the table item control style.

## RenderContents

[C#] protected override void RenderContents(HtmlTextWriter writer);  
[C++] protected: void RenderContents(HtmlTextWriter\* writer);  
[VB] Overrides Protected Sub RenderContents(ByVal writer As HtmlTextWriter)  
[JScript] protected override function RenderContents(writer : HtmlTextWriter);

### *Description*

1 A protected method. The output stream that renders HTML content to the  
2 client.

3 TableCellCollection class (System.Web.UI.WebControls)

4 TrackViewState

5  
6  
7 *Description*

8 Encapsulates a collection of  
9 **System.Web.UI.WebControls.TableHeaderCell** and  
10 **System.Web.UI.WebControls.TableCell** objects that make up a row in a  
11 **System.Web.UI.WebControls.Table** control. This class cannot be inherited.

12 Use this class to programmatically manage a collection of  
13 **System.Web.UI.WebControls.TableCell** objects that make up a row in a  
14 **System.Web.UI.WebControls.Table** control. This class is commonly used to add  
15 or remove cells from a row in a **System.Web.UI.WebControls.Table** control.

16 Count

17 TrackViewState

18  
19 [C#] public int Count {get;}

20 [C++] public: \_\_property int get\_Count();

21 [VB] Public ReadOnly Property Count As Integer

22 [JScript] public function get Count() : int;

23  
24 *Description*  
25

1 Gets the number of **System.Web.UI.WebControls.TableCell** objects in  
2 the **System.Web.UI.WebControls.TableCellCollection** .

3 Use this property to determine the number of cells in the  
4 **System.Web.UI.WebControls.TableCellCollection** . The  
5 **System.Web.UI.WebControls.TableCellCollection.Count** property is often  
6 used when iterating through the collection to determine the upper bound.

7 **IsReadOnly**

8 **TrackViewState**

9  
10 [C#] public bool IsReadOnly {get;}

11 [C++] public: \_\_property bool get\_IsReadOnly();

12 [VB] Public ReadOnly Property IsReadOnly As Boolean

13 [JScript] public function get IsReadOnly() : Boolean;

14  
15 *Description*

16 Gets a value indicating whether the  
17 **System.Web.UI.WebControls.TableCellCollection** is read-only.

18 This property always returns **false** to indicate that the  
19 **System.Web.UI.WebControls.TableCellCollection** can be written to in all cases.

20 **IsSynchronized**

21 **TrackViewState**

22  
23 [C#] public bool IsSynchronized {get;}

24 [C++] public: \_\_property bool get\_IsSynchronized();

25 [VB] Public ReadOnly Property IsSynchronized As Boolean

1 [JScript] public function get IsSynchronized() : Boolean;

3 *Description*

4 Gets a value indicating whether access to the  
5 **System.Web.UI.WebControls.TableCellCollection** is synchronized (thread-  
6 safe).

7 This property is derived from **System.Collections.ICollection** and is  
8 overridden to always return **false** .

9 Item

10 TrackViewState

12 [C#] public TableCell this[int index] {get;}

13 [C++] public: \_\_property TableCell\* get\_Item(int index);

14 [VB] Public Default ReadOnly Property Item(ByVal index As Integer) As  
15 TableCell

16 [JScript] returnValue = TableCellCollectionObject.Item(index);

18 *Description*

19 Gets a **System.Web.UI.WebControls.TableCell** from the  
20 **System.Web.UI.WebControls.TableCellCollection** at the specified index.

21 Use this indexer to get an individual  
22 **System.Web.UI.WebControls.TableCell** in the  
23 **System.Web.UI.WebControls.TableCellCollection** at the specified index using  
24 simple array notation. An ordinal index value that specifies the  
25 **System.Web.UI.WebControls.TableCell** to return.

1        SyncRoot

2        TrackViewState

3  
4    [C#] public object SyncRoot {get;}

5    [C++] public: \_\_property Object\* get\_SyncRoot();

6    [VB] Public ReadOnly Property SyncRoot As Object

7    [JScript] public function get SyncRoot() : Object;

8  
9    *Description*

10       Gets the object that can be used to synchronize access to the  
11       **System.Web.UI.WebControls.TableCellCollection** .

12       The object returned in this implementation is the  
13       **System.Web.UI.WebControls.TableCellCollection** object.

14       Add

15  
16    [C#] public int Add(TableCell cell);

17    [C++] public: int Add(TableCell\* cell);

18    [VB] Public Function Add(ByVal cell As TableCell) As Integer

19    [JScript] public function Add(cell : TableCell) : int;

20  
21    *Description*

22       Appends the specified **System.Web.UI.WebControls.TableCell** to the end  
23       of the **System.Web.UI.WebControls.TableCellCollection** .

24       Use this method to add the specified  
25       **System.Web.UI.WebControls.TableCell** to the end of a

1 **System.Web.UI.WebControls.TableCellCollection** . The  
2 **System.Web.UI.WebControls.TableCell** to add to the collection.

### 3 AddAt

4  
5 [C#] public void AddAt(int index, TableCell cell);  
6 [C++] public: void AddAt(int index, TableCell\* cell);  
7 [VB] Public Sub AddAt(ByVal index As Integer, ByVal cell As TableCell)  
8 [JScript] public function AddAt(index : int, cell : TableCell);  
9

### 10 *Description*

11 Adds the specified **System.Web.UI.WebControls.TableCell** to the  
12 **System.Web.UI.WebControls.TableCellCollection** at the specified index  
13 location.

14 Use this method to insert the specified  
15 **System.Web.UI.WebControls.TableCell** in a  
16 **System.Web.UI.WebControls.TableCellCollection** at the specified index. The  
17 location in the **System.Web.UI.WebControls.TableCellCollection** at which to  
18 add the **System.Web.UI.WebControls.TableCell**. The  
19 **System.Web.UI.WebControls.TableCell** to add to the  
20 **System.Web.UI.WebControls.TableCellCollection**.

### 21 AddRange

22  
23 [C#] public void AddRange(TableCell[] cells);  
24 [C++] public: void AddRange(TableCell\* cells[]);  
25 [VB] Public Sub AddRange(ByVal cells() As TableCell)

1 [JScript] public function AddRange(cells : TableCell[]);

2  
3 *Description*

4  
5 Clear

6  
7 [C#] public void Clear();

8 [C++] public: \_\_sealed void Clear();

9 [VB] NotOverridable Public Sub Clear()

10 [JScript] public function Clear(); Removes all

11 **System.Web.UI.WebControls.TableCell** objects from the

12 **System.Web.UI.WebControls.TableCellCollection** .

13  
14 *Description*

15 Removes all **System.Web.UI.WebControls.TableCell** objects from the

16 **System.Web.UI.WebControls.TableCellCollection** .

17 Use this method to remove all **System.Web.UI.WebControls.TableCell**  
18 objects from the **System.Web.UI.WebControls.TableCellCollection** and set the  
19 **System.Web.UI.WebControls.TableCellCollection.Count** property to 0.

20 CopyTo

21  
22 [C#] public void CopyTo(Array array, int index);

23 [C++] public: \_\_sealed void CopyTo(Array\* array, int index);

24 [VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As  
25 Integer)

1 [JScript] public function CopyTo(array : Array, index : int);

3 *Description*

4 Copies the items from the  
5 **System.Web.UI.WebControls.TableCellCollection** to the specified  
6 **System.Array** , starting with the specified index in the **System.Array** .

7 Use this method to copy the contents of the  
8 **System.Web.UI.WebControls.TableCellCollection** into the specified  
9 **System.Array** starting at the specified index. A zero-based **System.Array** that  
10 receives the copied items from the  
11 **System.Web.UI.WebControls.TableCellCollection**. The first index in the  
12 specified **System.Array** to receive the items.

13 **GetCellIndex**

15 [C#] public int GetCellIndex(TableCell cell);

16 [C++] public: int GetCellIndex(TableCell\* cell);

17 [VB] Public Function GetCellIndex(ByVal cell As TableCell) As Integer

18 [JScript] public function GetCellIndex(cell : TableCell) : int;

20 *Description*

21 Returns a value that represents the index of the specified  
22 **System.Web.UI.WebControls.TableCell** from the  
23 **System.Web.UI.WebControls.TableCellCollection** .

24 *Return Value:* The index of the specified

25 **System.Web.UI.WebControls.TableCell** within the



**System.Web.UI.WebControls.TableCellCollection** . The default is -1, which indicates that a match has not been found.

Use this method to determine the index of the specified **System.Web.UI.WebControls.TableCell** in the **System.Web.UI.WebControls.TableCellCollection** . If the specified **System.Web.UI.WebControls.TableCell** is not found, an index of -1 is returned. The **System.Web.UI.WebControls.TableCell** to get the index of in the **System.Web.UI.WebControls.TableCellCollection**.

GetEnumerator

[C#] public IEnumerator GetEnumerator();

[C++] public: \_\_sealed IEnumerator\* GetEnumerator();

[VB] NotOverridable Public Function GetEnumerator() As IEnumerator

[JScript] public function GetEnumerator() : IEnumerator;

### *Description*

Returns a **System.Collections.IEnumerator** that contains all **System.Web.UI.WebControls.TableCell** objects in the **System.Web.UI.WebControls.TableCellCollection** .

*Return Value:* A **System.Collections.IEnumerator** that contains all **System.Web.UI.WebControls.TableCell** objects within the **System.Web.UI.WebControls.TableCellCollection** .

Use this method to create a **System.Collections.IEnumerator** that can be iterated through easily to get each item in the **System.Web.UI.WebControls.TableCellCollection** .

## Remove

```
[C#] public void Remove(TableCell cell);  
[C++] public: void Remove(TableCell* cell);  
[VB] Public Sub Remove(ByVal cell As TableCell)  
[JScript] public function Remove(cell : TableCell);
```

### *Description*

Removes the specified **System.Web.UI.WebControls.TableCell** from the **System.Web.UI.WebControls.TableCellCollection**.

Use this method to remove the specified **System.Web.UI.WebControls.TableCell** from a **System.Web.UI.WebControls.TableCellCollection**. The **System.Web.UI.WebControls.TableCell** to remove from the **System.Web.UI.WebControls.TableCellCollection**.

## RemoveAt

```
[C#] public void RemoveAt(int index);  
[C++] public: __sealed void RemoveAt(int index);  
[VB] NotOverridable Public Sub RemoveAt(ByVal index As Integer)  
[JScript] public function RemoveAt(index : int);
```

### *Description*

Removes a **System.Web.UI.WebControls.TableCell** from the **System.Web.UI.WebControls.TableCellCollection** at the specified index.

1        Use this method to remove a **System.Web.UI.WebControls.TableCell**  
2 from a **System.Web.UI.WebControls.TableCellCollection** at the specified  
3 index. The index of the **System.Web.UI.WebControls.TableCell** to remove from  
4 the **System.Web.UI.WebControls.TableCellCollection**.

5        **IList.Add**

6  
7 [C#] int IList.Add(object o);

8 [C++] int IList::Add(Object\* o);

9 [VB] Function Add(ByVal o As Object) As Integer Implements IList.Add

10 [JScript] function IList.Add(o : Object) : int;

11        **IList.Contains**

12  
13 [C#] bool IList.Contains(object o);

14 [C++] bool IList::Contains(Object\* o);

15 [VB] Function Contains(ByVal o As Object) As Boolean Implements

16 **IList.Contains**

17 [JScript] function IList.Contains(o : Object) : Boolean;

18        **IList.IndexOf**

19  
20 [C#] int IList.IndexOf(object o);

21 [C++] int IList::IndexOf(Object\* o);

22 [VB] Function IndexOf(ByVal o As Object) As Integer Implements IList.IndexOf

23 [JScript] function IList.IndexOf(o : Object) : int;

24        **IList.Insert**

25

1  
2 [C#] void IList.Insert(int index, object o);

3 [C++] void IList::Insert(int index, Object\* o);

4 [VB] Sub Insert(ByVal index As Integer, ByVal o As Object) Implements

5 IList.Insert

6 [JScript] function IList.Insert(index : int, o : Object);

7       IList.Remove

8  
9 [C#] void IList.Remove(object o);

10 [C++] void IList::Remove(Object\* o);

11 [VB] Sub Remove(ByVal o As Object) Implements IList.Remove

12 [JScript] function IList.Remove(o : Object);

13       TableCellControlBuilder class (System.Web.UI.WebControls)

14       ToString

15  
16  
17 *Description*

18       Interacts with the parser to build a

19 **System.Web.UI.WebControls.TableCell** control.

20       To create a custom control builder for a

21 **System.Web.UI.WebControls.TableCell** derived control, you need to inherit

22 from this class.

23       TableCellControlBuilder

24 *Example Syntax:*

25       ToString

```

1
2 [C#] public TableCellControlBuilder();
3 [C++] public: TableCellControlBuilder();
4 [VB] Public Sub New()
5 [JScript] public function TableCellControlBuilder();
6     ControlType
7     FChildrenAsProperties
8     FIsNonParserAccessor
9     HasAspCode
10    ID
11    InDesigner
12    NamingContainerType
13    Parser
14    TagName
15    AllowWhitespaceLiterals
16
17 [C#] public override bool AllowWhitespaceLiterals();
18 [C++] public: bool AllowWhitespaceLiterals();
19 [VB] Overrides Public Function AllowWhitespaceLiterals() As Boolean
20 [JScript] public override function AllowWhitespaceLiterals() : Boolean;
21

```

## *Description*

Specifies whether white space literals are allowed.

*Return Value:* **false** for all cases.

1 This method overrides  
2 **System.Web.UI.ControlBuilder.AllowWhitespaceLiterals** to ignore white  
3 space in the **System.Web.UI.WebControls.TableCell** control.

4 TableHeaderCell class (System.Web.UI.WebControls)  
5 ToString

6  
7  
8 *Description*

9 Represents a heading cell within a **System.Web.UI.WebControls.Table**  
10 control.

11 The **System.Web.UI.WebControls.TableHeaderCell** class represents a  
12 heading cell in a **System.Web.UI.WebControls.Table** control. You can use the  
13 **System.Web.UI.WebControls.TableCell.Text** property to specify or determine  
14 the contents of the heading cell.

15 TableHeaderCell

16 *Example Syntax:*

17 ToString

18  
19 [C#] public TableHeaderCell();  
20 [C++] public: TableHeaderCell();  
21 [VB] Public Sub New()  
22 [JScript] public function TableHeaderCell();  
23

24 *Description*  
25

1        Initializes a new instance of the  
2        **System.Web.UI.WebControls.TableHeaderCell** class.  
3        Use this constructor to create and initialize a new instance of the  
4        **System.Web.UI.WebControls.TableHeaderCell** class.

5        AccessKey

6        Attributes

7        BackColor

8        BorderColor

9        BorderStyle

10       BorderWidth

11       ChildControlsCreated

12       ClientID

13       ColumnSpan

14       Context

15       Controls

16       ControlStyle

17       ControlStyleCreated

18       CssClass

19       Enabled

20       EnableViewState

21       Events

22       Font

23       ForeColor

24       HasChildViewState

25       Height

1	HorizontalAlign
2	ID
3	IsTrackingViewState
4	NamingContainer
5	Page
6	Parent
7	RowSpan
8	Site
9	Style
10	TabIndex
11	TagKey
12	TagName
13	TemplateSourceDirectory
14	Text
15	ToolTip
16	UniqueID
17	VerticalAlign
18	ViewState
19	ViewStateIgnoresCase
20	Visible
21	Width
22	Wrap
23	TableItemStyle class (System.Web.UI.WebControls)
24	TrackViewState
25	



1  
2  
3 *Description*

4 Specifies the style for an item in a table control.

5 The **System.Web.UI.WebControls.TableItemStyle** class represents the  
6 style properties for an item in the **System.Web.UI.WebControls.Table** control.

7 You can control the a Specifies the style of the table item.

8 **TableItemStyle**

9 *Example Syntax:*

10 **TrackViewState**

11  
12 [C#] public TableItemStyle();

13 [C++] public: TableItemStyle();

14 [VB] Public Sub New()

15 [JScript] public function TableItemStyle(); Creates a new instance of the

16 **System.Web.UI.WebControls.TableItemStyle** class.

17  
18 *Description*

19 Creates a new instance of the

20 **System.Web.UI.WebControls.TableItemStyle** class.

21 **TableItemStyle**

22 *Example Syntax:*

23 **TrackViewState**

24  
25 [C#] public TableItemStyle(StateBag bag);

```

1 [C++] public: TableItemStyle(StateBag* bag);
2 [VB] Public Sub New(ByVal bag As StateBag)
3 [JScript] public function TableItemStyle(bag : StateBag);

```

#### *Description*

Creates a new instance of the **System.Web.UI.WebControls.TableItemStyle** class with the specified state bag. The specified state bag.

BackColor  
 BorderColor  
 BorderStyle  
 BorderWidth  
 Container  
 CssClass  
 DesignMode  
 Events  
 Font  
 ForeColor  
 Height  
 HorizontalAlign  
 TrackViewState

#### *Description*

Gets or sets the horizontal alignment of the cell content.

1 IsEmpty  
2 IsTrackingViewState  
3 Site  
4 VerticalAlign  
5 TrackViewState  
6  
7

8 *Description*

9 Gets or sets the vertical alignment of the cell content.

10 ViewState  
11 Width  
12 Wrap  
13 TrackViewState  
14  
15

16 *Description*

17 Gets or sets a value indicating whether the cell content wraps within the  
18 cell.

19 AddAttributesToRender  
20

21 [C#] public override void AddAttributesToRender(HtmlTextWriter writer,  
22 WebControl owner);

23 [C++] public: void AddAttributesToRender(HtmlTextWriter\* writer,  
24 WebControl\* owner);

25 [VB] Overrides Public Sub AddAttributesToRender(ByVal writer As

1 HtmlTextWriter, ByVal owner As WebControl)

2 [JScript] public override function AddAttributesToRender(writer :

3 HtmlTextWriter, owner : WebControl);

4  
5 *Description*

6 Adds information about horizontal alignment, vertical alignment, and wrap  
7 to the list of attributes to render. The output stream that renders HTML content to  
8 the client. The control that the style refers to.

9 CopyFrom

10  
11 [C#] public override void CopyFrom(Style s);

12 [C++] public: void CopyFrom(Style\* s);

13 [VB] Overrides Public Sub CopyFrom(ByVal s As Style)

14 [JScript] public override function CopyFrom(s : Style);

15  
16 *Description*

17 Copies non-blank elements from the specified style, overwriting existing  
18 style elements if necessary. The style to copy.

19 MergeWith

20  
21 [C#] public override void MergeWith(Style s);

22 [C++] public: void MergeWith(Style\* s);

23 [VB] Overrides Public Sub MergeWith(ByVal s As Style)

24 [JScript] public override function MergeWith(s : Style);

1  
2 *Description*

3 Copies non-blank elements from the specified style, but will not overwrite  
4 any existing style elements. The style to copy.

5 Reset

6  
7 [C#] public override void Reset();

8 [C++] public: void Reset();

9 [VB] Overrides Public Sub Reset()

10 [JScript] public override function Reset();  
11

12 *Description*

13 Clears out any defined style elements from the state bag.

14 TableRow class (System.Web.UI.WebControls)

15 TrackViewState  
16  
17

18 *Description*

19 Represents a row in a **System.Web.UI.WebControls.Table** control.

20 The **System.Web.UI.WebControls.TableRow** class represents a row in a  
21 **System.Web.UI.WebControls.Table** control.

22 TableRow

23 *Example Syntax:*

24 TrackViewState  
25

```

1
2 [C#] public TableRow();
3 [C++] public: TableRow();
4 [VB] Public Sub New()
5 [JScript] public function TableRow();
6

```

### *Description*

Initializes a new instance of the **System.Web.UI.WebControls.TableRow** class.

Use this constructor to create and initialize a new instance of the **System.Web.UI.WebControls.TableRow** class.

AccessKey

Attributes

BackColor

BorderColor

BorderStyle

BorderWidth

Cells

TrackViewState

### *Description*

Gets a collection of **System.Web.UI.WebControls.TableCell** objects that represent the cells of a row in a **System.Web.UI.WebControls.Table** control.

Use this property to programmatically control a collection of **System.Web.UI.WebControls.TableCell** objects that represent the cells from a row of the **System.Web.UI.WebControls.Table** control. You can programmatically add, insert, or remove a **System.Web.UI.WebControls.TableCell** object from the collection.

ChildControlsCreated

ClientID

Context

Controls

ControlStyle

ControlStyleCreated

CssClass

Enabled

EnableViewState

Events

Font

ForeColor

HasChildViewState

Height

HorizontalAlign

TrackViewState

#### *Description*

Gets or sets the horizontal alignment of the contents in the row.

1 Use the **System.Web.UI.WebControls.TableRow.HorizontalAlign**  
2 property to specify the horizontal alignment of the contents of the row. The  
3 following table lists the possible values.

4 ID  
5 IsTrackingViewState  
6 NamingContainer  
7 Page  
8 Parent  
9 Site  
10 Style  
11 TabIndex  
12 TagKey  
13 TagName  
14 TemplateSourceDirectory  
15 ToolTip  
16 UniqueID  
17 VerticalAlign  
18 TrackViewState

19  
20  
21 *Description*

22 Gets or sets the vertical alignment of the contents in the row.

23 Use the **System.Web.UI.WebControls.TableRow.VerticalAlign** property  
24 to specify the vertical alignment of the contents of the row. The following table  
25 lists the possible values.



1 ViewState  
2 ViewStateIgnoresCase  
3 Visible  
4 Width  
5 CreateControlCollection

6  
7 [C#] protected override ControlCollection CreateControlCollection();  
8 [C++] protected: ControlCollection\* CreateControlCollection();  
9 [VB] Overrides Protected Function CreateControlCollection() As  
10 ControlCollection  
11 [JScript] protected override function CreateControlCollection() :  
12 ControlCollection;

13  
14 *Description*

15  
16 CreateControlStyle

17  
18 [C#] protected override Style CreateControlStyle();  
19 [C++] protected: Style\* CreateControlStyle();  
20 [VB] Overrides Protected Function CreateControlStyle() As Style  
21 [JScript] protected override function CreateControlStyle() : Style;

22  
23 *Description*  
24  
25

1 A protected method. Creates a table item control style.  
2 *Return Value:* A **System.Web.UI.WebControls.Style** that specifies the table item  
3 control style.

4 TableRowCollection class (System.Web.UI.WebControls)

5 TrackViewState

6  
7  
8 *Description*

9 Encapsulates a collection of **System.Web.UI.WebControls.TableRow**  
10 objects that represent a single row in a **System.Web.UI.WebControls.Table**  
11 control. This class cannot be inherited.

12 Use this class to programmatically manage a collection of  
13 **System.Web.UI.WebControls.TableRow** objects. This class is commonly used  
14 to add or remove rows from a **System.Web.UI.WebControls.Table** control.

15 Count

16 TrackViewState

17  
18 [C#] public int Count {get;}

19 [C++] public: \_\_property int get\_Count();

20 [VB] Public ReadOnly Property Count As Integer

21 [JScript] public function get Count() : int;

22  
23 *Description*

24 Gets the number of **System.Web.UI.WebControls.TableRow** objects in  
25 the **System.Web.UI.WebControls.TableRowCollection** .

1        Use this property to determine the number of rows in the  
2        **System.Web.UI.WebControls.TableRowCollection** . The  
3        **System.Web.UI.WebControls.TableRowCollection.Count** property is often  
4        used when iterating through the collection to determine the upper bound.

5        **IsReadOnly**

6        **TrackViewState**

7  
8        [C#] public bool IsReadOnly {get;}

9        [C++] public: \_\_property bool get\_IsReadOnly();

10        [VB] Public ReadOnly Property IsReadOnly As Boolean

11        [JScript] public function get IsReadOnly() : Boolean;

12  
13        *Description*

14        Gets a value indicating whether the  
15        **System.Web.UI.WebControls.TableRowCollection** is read-only.

16        This property always returns **false** to indicate that the  
17        **System.Web.UI.WebControls.TableRowCollection** can be written to in all  
18        cases.

19        **IsSynchronized**

20        **TrackViewState**

21  
22        [C#] public bool IsSynchronized {get;}

23        [C++] public: \_\_property bool get\_IsSynchronized();

24        [VB] Public ReadOnly Property IsSynchronized As Boolean

25        [JScript] public function get IsSynchronized() : Boolean;

1  
2 *Description*

3 Gets a value indicating whether access to the  
4 **System.Web.UI.WebControls.TableRowCollection** is synchronized (thread-  
5 safe).

6 This property is derived from **System.Collections.ICollection** and is  
7 overridden to always return **false** .

8 Item

9 TrackViewState

10  
11 [C#] public TableRow this[int index] {get;}

12 [C++] public: \_\_property TableRow\* get\_Item(int index);

13 [VB] Public Default ReadOnly Property Item(ByVal index As Integer) As  
14 TableRow

15 [JScript] returnValue = TableRowCollectionObject.Item(index);  
16

17 *Description*

18 Gets a **System.Web.UI.WebControls.TableRow** from the  
19 **System.Web.UI.WebControls.TableRowCollection** at the specified index.

20 Use this indexer to get an individual  
21 **System.Web.UI.WebControls.TableRow** from the  
22 **System.Web.UI.WebControls.TableRowCollection** at the specified index using  
23 simple array notation. An ordinal index value that specifies which  
24 **System.Web.UI.WebControls.TableRow** to return.

25 SyncRoot

## TrackViewState

[C#] public object SyncRoot {get;}

[C++] public: \_\_property Object\* get\_SyncRoot();

[VB] Public ReadOnly Property SyncRoot As Object

[JScript] public function get SyncRoot() : Object;

### *Description*

Gets the object that can be used to synchronize access to the **System.Web.UI.WebControls.TableRowCollection**.

The object returned in this implementation is the **System.Web.UI.WebControls.TableRowCollection** object itself.

### Add

[C#] public int Add(TableRow row);

[C++] public: int Add(TableRow\* row);

[VB] Public Function Add(ByVal row As TableRow) As Integer

[JScript] public function Add(row : TableRow) : int;

### *Description*

Appends the specified **System.Web.UI.WebControls.TableRow** to the end of the **System.Web.UI.WebControls.TableRowCollection**.

Use this method to add a **System.Web.UI.WebControls.TableRow** to the end of a **System.Web.UI.WebControls.TableRowCollection**. The

1 **System.Web.UI.WebControls.TableRow** to add to the

2 **System.Web.UI.WebControls.TableRowCollection**.

3       AddAt

4  
5 [C#] public void AddAt(int index, TableRow row);

6 [C++] public: void AddAt(int index, TableRow\* row);

7 [VB] Public Sub AddAt(ByVal index As Integer, ByVal row As TableRow)

8 [JScript] public function AddAt(index : int, row : TableRow);

9  
10 *Description*

11       Adds the specified **System.Web.UI.WebControls.TableRow** to the  
12 **System.Web.UI.WebControls.TableRowCollection** at the specified index  
13 location.

14       Use this method to insert the specified  
15 **System.Web.UI.WebControls.TableRow** in a  
16 **System.Web.UI.WebControls.TableRowCollection** at the specified index. The  
17 location in the **System.Web.UI.WebControls.TableRowCollection** at which to  
18 add the **System.Web.UI.WebControls.TableRow**. The  
19 **System.Web.UI.WebControls.TableRow** to add to the  
20 **System.Web.UI.WebControls.TableRowCollection**.

21       AddRange

22  
23 [C#] public void AddRange(TableRow[] rows);

24 [C++] public: void AddRange(TableRow\* rows[]);

25 [VB] Public Sub AddRange(ByVal rows() As TableRow)

1 [JScript] public function AddRange(rows : TableRow[]);

2  
3 *Description*

4  
5 Clear

6  
7 [C#] public void Clear();

8 [C++] public: \_\_sealed void Clear();

9 [VB] NotOverridable Public Sub Clear()

10 [JScript] public function Clear();

11  
12 *Description*

13 Removes all **System.Web.UI.WebControls.TableRow** controls from the  
14 **System.Web.UI.WebControls.TableRowCollection** .

15 Use this method to remove all **System.Web.UI.WebControls.TableRow**  
16 objects from the **System.Web.UI.WebControls.TableRowCollection** and set the  
17 **System.Web.UI.WebControls.TableRowCollection.Count** property to 0.

18 CopyTo

19  
20 [C#] public void CopyTo(Array array, int index);

21 [C++] public: \_\_sealed void CopyTo(Array\* array, int index);

22 [VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As  
23 Integer)

24 [JScript] public function CopyTo(array : Array, index : int);

1  
2 *Description*

3 Copies the items from the  
4 **System.Web.UI.WebControls.TableRowCollection** to the specified  
5 **System.Array** , starting with the specified index in the **System.Array** .

6 Use this method to copy the contents of the  
7 **System.Web.UI.WebControls.TableRowCollection** into the specified  
8 **System.Array** , starting at the specified index. A zero-based **System.Array** that  
9 receives the copied items from the  
10 **System.Web.UI.WebControls.TableRowCollection**. The first position in the  
11 specified **System.Array** to receive copied contents.

12 **GetEnumerator**

13  
14 [C#] public IEnumerator GetEnumerator();  
15 [C++] public: \_\_sealed IEnumerator\* GetEnumerator();  
16 [VB] NotOverridable Public Function GetEnumerator() As IEnumerator  
17 [JScript] public function GetEnumerator() : IEnumerator;  
18

19 *Description*

20 Returns an **System.Collections.IEnumerator** that contains all  
21 **System.Web.UI.WebControls.TableRow** objects within the  
22 **System.Web.UI.WebControls.TableRowCollection** .

23 *Return Value:* A **System.Collections.IEnumerator** that contains all  
24 **System.Web.UI.WebControls.TableRow** objects within the  
25 **System.Web.UI.WebControls.TableRowCollection** .



1 Use this method to create a **System.Collections.IEnumerator** that can be  
2 iterated through easily to get each item in the

3 **System.Web.UI.WebControls.TableRowCollection** .

4 **GetRowIndex**

6 [C#] public int GetRowIndex(TableRow row);

7 [C++] public: int GetRowIndex(TableRow\* row);

8 [VB] Public Function GetRowIndex(ByVal row As TableRow) As Integer

9 [JScript] public function GetRowIndex(row : TableRow) : int;

11 *Description*

12 Returns a value that represents the index of the specified

13 **System.Web.UI.WebControls.TableRow** from the

14 **System.Web.UI.WebControls.TableRowCollection** .

15 *Return Value:* The ordinal index position of the specified

16 **System.Web.UI.WebControls.TableRow** within the collection. The default is -1,

17 which indicates that the specified **System.Web.UI.WebControls.TableRow** has

18 not been found.

19 Use this method to determine the index of the specified

20 **System.Web.UI.WebControls.TableRow** in the

21 **System.Web.UI.WebControls.TableRowCollection** . If the specified

22 **System.Web.UI.WebControls.TableRow** is not found, an index of -1 is returned.

23 The **System.Web.UI.WebControls.TableRow** to search for in the

24 **System.Web.UI.WebControls.TableRowCollection**.

25 **Remove**

```

1
2 [C#] public void Remove(TableRow row);
3 [C++] public: void Remove(TableRow* row);
4 [VB] Public Sub Remove(ByVal row As TableRow)
5 [JScript] public function Remove(row : TableRow);
6

```

### *Description*

Removes the specified **System.Web.UI.WebControls.TableRow** from the **System.Web.UI.WebControls.TableRowCollection**.

Use this method to remove the specified **System.Web.UI.WebControls.TableRow** from a **System.Web.UI.WebControls.TableRowCollection**. The **System.Web.UI.WebControls.TableRow** to remove from the **System.Web.UI.WebControls.TableRowCollection**.

### **RemoveAt**

```

17 [C#] public void RemoveAt(int index);
18 [C++] public: __sealed void RemoveAt(int index);
19 [VB] NotOverridable Public Sub RemoveAt(ByVal index As Integer)
20 [JScript] public function RemoveAt(index : int);
21

```

### *Description*

Removes a **System.Web.UI.WebControls.TableRow** from the **System.Web.UI.WebControls.TableRowCollection** at the specified index.

1 Use this method to remove a **System.Web.UI.WebControls.TableRow**  
2 from the **System.Web.UI.WebControls.TableRowCollection** at the specified  
3 index. The index of the **System.Web.UI.WebControls.TableRow** to remove  
4 from the **System.Web.UI.WebControls.TableRowCollection**.

5 **IList.Add**

6  
7 [C#] int IList.Add(object o);

8 [C++] int IList::Add(Object\* o);

9 [VB] Function Add(ByVal o As Object) As Integer Implements IList.Add

10 [JScript] function IList.Add(o : Object) : int;

11 **IList.Contains**

12  
13 [C#] bool IList.Contains(object o);

14 [C++] bool IList::Contains(Object\* o);

15 [VB] Function Contains(ByVal o As Object) As Boolean Implements

16 **IList.Contains**

17 [JScript] function IList.Contains(o : Object) : Boolean;

18 **IList.IndexOf**

19  
20 [C#] int IList.IndexOf(object o);

21 [C++] int IList::IndexOf(Object\* o);

22 [VB] Function IndexOf(ByVal o As Object) As Integer Implements IList.IndexOf

23 [JScript] function IList.IndexOf(o : Object) : int;

24 **IList.Insert**

25

1  
2 [C#] void IList.Insert(int index, object o);  
3 [C++] void IList::Insert(int index, Object\* o);  
4 [VB] Sub Insert(ByVal index As Integer, ByVal o As Object) Implements  
5 IList.Insert  
6 [JScript] function IList.Insert(index : int, o : Object);

7       IList.Remove

8  
9 [C#] void IList.Remove(object o);  
10 [C++] void IList::Remove(Object\* o);  
11 [VB] Sub Remove(ByVal o As Object) Implements IList.Remove  
12 [JScript] function IList.Remove(o : Object);

13       TableStyle class (System.Web.UI.WebControls)

14       ToString

15  
16  
17 *Description*

18       Specifies the style for a table control.

19       This is used primarily by control developers.

20       TableStyle

21 *Example Syntax:*

22       ToString

23  
24 [C#] public TableStyle();

25 [C++] public: TableStyle();

1 [VB] Public Sub New()  
2 [JScript] public function TableStyle(); Initializes a new instance of the  
3 **System.Web.UI.WebControls.TableStyle** class.

4  
5 *Description*

6       Initializes a new instance of the **System.Web.UI.WebControls.TableStyle**  
7 class using default values.

8       TableStyle

9       *Example Syntax:*

10       ToString

11  
12 [C#] public TableStyle(StateBag bag);

13 [C++] public: TableStyle(StateBag\* bag);

14 [VB] Public Sub New(ByVal bag As StateBag)

15 [JScript] public function TableStyle(bag : StateBag);

16  
17 *Description*

18       Initializes a new instance of the **System.Web.UI.WebControls.TableStyle**  
19 class with the specified state bag information.

20       BackColor

21       BackImageUrl

22       ToString

23  
24  
25 *Description*

1 Gets or sets the URL of an image to display in the background of a table  
2 control.

3 The image will be tiled if it is smaller than the table.

4 BorderColor

5 BorderStyle

6 BorderWidth

7 CellPadding

8 ToString

9  
10  
11 *Description*

12 Gets or sets the distance between the border and the contents of the table  
13 cell.

14 CellSpacing

15 ToString

16  
17 [C#] public virtual int CellSpacing {get; set;}

18 [C++] public: \_\_property virtual int get\_CellSpacing();public: \_\_property virtual

19 void set\_CellSpacing(int);

20 [VB] Overridable Public Property CellSpacing As Integer

21 [JScript] public function get CellSpacing() : int;public function set

22 CellSpacing(int);

23  
24 *Description*

25 Gets or sets the distance between table cells.

1	Container
2	CssClass
3	DesignMode
4	Events
5	Font
6	ForeColor
7	GridLines
8	ToString
9	
10	

*Description*

12	Gets or sets the gridlines property of the table.
13	Height
14	HorizontalAlign
15	ToString
16	
17	

*Description*

19	Gets or sets the horizontal alignment of the table within the page.
20	IsEmpty
21	IsTrackingViewState
22	Site
23	ViewState
24	Width
25	AddAttributesToRender

1  
2 [C#] public override void AddAttributesToRender(HtmlTextWriter writer,  
3 WebControl owner);

4 [C++] public: void AddAttributesToRender(HtmlTextWriter\* writer,  
5 WebControl\* owner);

6 [VB] Overrides Public Sub AddAttributesToRender(ByVal writer As  
7 HtmlTextWriter, ByVal owner As WebControl)

8 [JScript] public override function AddAttributesToRender(writer :  
9 HtmlTextWriter, owner : WebControl);

10  
11 *Description*

12 Adds information about the background image, callspacing, cellpadding,  
13 gridlines, and alignment to the list of attributes to render. The output stream that  
14 renders HTML content to the client. The control associated with the style.

15 CopyFrom

16  
17 [C#] public override void CopyFrom(Style s);

18 [C++] public: void CopyFrom(Style\* s);

19 [VB] Overrides Public Sub CopyFrom(ByVal s As Style)

20 [JScript] public override function CopyFrom(s : Style);

21  
22 *Description*

23 Copies non-blank elements from the specified style, overwriting existing  
24 style elements if necessary. The style to copy.

25 MergeWith



1  
2 [C#] public override void MergeWith(Style s);  
3 [C++] public: void MergeWith(Style\* s);  
4 [VB] Overrides Public Sub MergeWith(ByVal s As Style)  
5 [JScript] public override function MergeWith(s : Style);  
6

7 *Description*

8 Copies non-blank elements from the specified style, but will not overwrite  
9 any existing style elements. The style to copy.

10 Reset

11  
12 [C#] public override void Reset();  
13 [C++] public: void Reset();  
14 [VB] Overrides Public Sub Reset()  
15 [JScript] public override function Reset();  
16

17 *Description*

18 Clears out any defined style elements from the state bag.

19 TargetConverter class (System.Web.UI.WebControls)

20 TrackViewState

21  
22  
23 *Description*

24 TargetConverter

25 *Example Syntax:*

## TrackViewState

[C#] public TargetConverter();

[C++] public: TargetConverter();

[VB] Public Sub New()

[JScript] public function TargetConverter();

## GetStandardValues

[C#] public override StandardValuesCollection

GetStandardValues(ITypeDescriptorContext context);

[C++] public: StandardValuesCollection\*

GetStandardValues(ITypeDescriptorContext\* context);

[VB] Overrides Public Function GetStandardValues(ByVal context As  
ITypeDescriptorContext) As StandardValuesCollection

[JScript] public override function GetStandardValues(context :

ITypeDescriptorContext) : StandardValuesCollection;

## *Description*

### GetStandardValuesExclusive

[C#] public override bool GetStandardValuesExclusive(ITypeDescriptorContext  
context);

[C++] public: bool GetStandardValuesExclusive(ITypeDescriptorContext\*  
context);

[VB] Overrides Public Function GetStandardValuesExclusive(ByVal context As

1 ITypeDescriptorContext) As Boolean

2 [JScript] public override function GetStandardValuesExclusive(context :

3 ITypeDescriptorContext) : Boolean;

4  
5 *Description*

6       GetStandardValuesSupported

7  
8 [C#] public override bool GetStandardValuesSupported(ITypeDescriptorContext  
9 context);

10 [C++] public: bool GetStandardValuesSupported(ITypeDescriptorContext\*  
11 context);

12 [VB] Overrides Public Function GetStandardValuesSupported(ByVal context As  
13 ITypeDescriptorContext) As Boolean

14 [JScript] public override function GetStandardValuesSupported(context :  
15 ITypeDescriptorContext) : Boolean;

16  
17 *Description*

18       TemplateColumn class (System.Web.UI.WebControls)

19       ToString

20  
21  
22 *Description*

23       A column type for the **System.Web.UI.WebControls.DataGrid** control  
24 that allows you to customize the layout of controls in the column.

1        Use the **System.Web.UI.WebControls.TemplateColumn** column type in  
2 a **System.Web.UI.WebControls.DataGrid** control to create a column with a  
3 customized control layout.

4        **TemplateColumn**

5        *Example Syntax:*

6        **ToString**

7  
8        [C#] public **TemplateColumn**();

9        [C++] public: **TemplateColumn**();

10        [VB] Public Sub **New**()

11        [JScript] public function **TemplateColumn**();

12  
13        *Description*

14        Initializes a new instance of the  
15 **System.Web.UI.WebControls.TemplateColumn** class.

16        Use this constructor to create and initializes a new instance of the  
17 **System.Web.UI.WebControls.TemplateColumn** class.

18        **DesignMode**

19        **EditItemTemplate**

20        **ToString**

21  
22  
23        *Description*

24        Gets or sets the template for this column for the item selected for editing in  
25 the **System.Web.UI.WebControls.DataGrid** control.

1 Use the  
2 **System.Web.UI.WebControls.TemplateColumn.EditItemTemplate** property to  
3 control the contents of the item selected for editing in the column of the  
4 **System.Web.UI.WebControls.DataGrid** control.

5 FooterStyle

6 FooterTemplate

7 ToString

8  
9  
10 *Description*

11 Gets or sets the template for the footer section of this column in the  
12 **System.Web.UI.WebControls.DataGrid** control.

13 Use the **System.Web.UI.WebControls.DataList.FooterTemplate**  
14 property to control the contents of the footer section.

15 FooterText

16 HeaderImageUrl

17 HeaderStyle

18 HeaderTemplate

19 ToString

20  
21  
22 *Description*

23 Gets or sets the template for the heading section of this column in the  
24 **System.Web.UI.WebControls.DataGrid** control.

1 Use the **System.Web.UI.WebControls.DataList.HeaderTemplate**  
2 property to control the contents of the heading section. The appearance of the  
3 header section is controlled by the  
4 **System.Web.UI.WebControls.DataList.HeaderStyle** property.

5 HeaderText

6 IsTrackingViewState

7 ItemStyle

8 ItemTemplate

9 ToString

10  
11  
12 *Description*

13 Gets or sets the template for the items in this column of the  
14 **System.Web.UI.WebControls.DataGrid** control.

15 Use the **System.Web.UI.WebControls.TemplateColumn.ItemTemplate**  
16 property to control the contents of the items in the  
17 **System.Web.UI.WebControls.DataList** control.

18 Owner

19 SortExpression

20 ViewState

21 Visible

22 InitializeCell

23  
24 [C#] public override void InitializeCell(TableCell cell, int columnIndex,  
25 ListItemType itemType);

```

1 [C++] public: void InitializeCell(TableCell* cell, int columnIndex, ListItemType
2 itemType);
3 [VB] Overrides Public Sub InitializeCell(ByVal cell As TableCell, ByVal
4 columnIndex As Integer, ByVal itemType As ListItemType)
5 [JScript] public override function InitializeCell(cell : TableCell, columnIndex : int,
6 itemType : ListItemType);

```

#### *Description*

TextAlign enumeration (System.Web.UI.WebControls)

TrackViewState

#### *Description*

Specifies whether the text associated with a checkbox or radio button control appears to the left or to the right of the control.

The **System.Web.UI.WebControls.TextAlign** enumeration represents the different text alignment options for checkbox and radio button controls.

TrackViewState

```

20 [C#] public const TextAlign Left;
21 [C++] public: const TextAlign Left;
22 [VB] Public Const Left As TextAlign
23 [JScript] public var Left : TextAlign;

```

#### *Description*

Text associated with a checkbox or radio button control appears to the left of the control.

TrackViewState

[C#] public const TextAlign Right;

[C++] public: const TextAlign Right;

[VB] Public Const Right As TextAlign

[JScript] public var Right : TextAlign;

#### *Description*

Text associated with a checkbox or radio button control appears to the right of the control.

TextBox class (System.Web.UI.WebControls)

ToString

#### *Description*

Constructs a text box and defines its properties.

The **System.Web.UI.WebControls.TextBox** server control is an input control that lets the user enter text. By default, the

**System.Web.UI.WebControls.TextBox.TextMode** of the text box is **SingleLine**, but it can be modified to be **MultiLine** or **Password**.

TextBox

*Example Syntax:*

ToString



```

1
2 [C#] public TextBox();
3 [C++] public: TextBox();
4 [VB] Public Sub New()
5 [JScript] public function TextBox();
6

```

### *Description*

Initializes a new instance of the **System.Web.UI.WebControls.TextBox** class.

AccessKey

Attributes

AutoPostBack

ToString

### *Description*

Gets or sets a value indicating whether an automatic postback to the server will occur whenever the user changes the content of the text box.

BackColor

BorderColor

BorderStyle

BorderWidth

ChildControlsCreated

ClientID

Columns

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

ToString

*Description*

Gets or sets the display width of the text box in characters.

Context

Controls

ControlStyle

ControlStyleCreated

CssClass

Enabled

EnableViewState

Events

Font

ForeColor

HasChildViewState

Height

ID

IsTrackingViewState

MaxLength

ToString

*Description*

Gets or sets the maximum number of characters allowed in the text box.

2025-06-06 14:20:00

1 This property is not applicable when the  
2 **System.Web.UI.WebControls.TextBox.TextMode** property is set to  
3 **TextBoxMode.MultiLine** .

4 NamingContainer

5 Page

6 Parent

7 ReadOnly

8 ToString

9  
10  
11 *Description*

12 Whether the textbox is in read-only mode.

13 Rows

14 ToString

15  
16 [C#] public virtual int Rows {get; set;}

17 [C++] public: \_\_property virtual int get\_Rows();public: \_\_property virtual void  
18 set\_Rows(int);

19 [VB] Overridable Public Property Rows As Integer

20 [JScript] public function get Rows() : int;public function set Rows(int);

21  
22 *Description*

23 Gets or sets the display height of a multiline text box.  
24  
25

This property is only applicable when the  
**System.Web.UI.WebControls.TextBox.TextMode** property is set to  
**TextBoxMode.MultiLine** .

Site

Style

TabIndex

TagKey

ToString

*Description*

A protected property. Gets the HTML tag for the text box control.

TagName

TemplateSourceDirectory

Text

ToString

*Description*

Gets or sets the text content of the text box.

TextMode

ToString

[C#] public virtual TextBoxMode TextMode {get; set;}

[C++] public: \_\_property virtual TextBoxMode get\_TextMode();public:

```

1  __property virtual void set_TextMode(TextBoxMode);
2  [VB] Overridable Public Property TextMode As TextBoxMode
3  [JScript] public function get TextMode() : TextBoxMode;public function set
4  TextMode(TextBoxMode);
5

```

#### 6 *Description*

7 Gets or sets the behavior mode of the text box.

8 ToolTip

9 UniqueID

10 ViewState

11 ViewStateIgnoresCase

12 Visible

13 Width

14 Wrap

15 ToString

#### 18 *Description*

19 Gets or sets a value indicating whether the text content wraps within the  
20 text box.

21 This property is only applicable when the

22 **System.Web.UI.WebControls.TextBox.TextMode** property is set to

23 **TextBoxMode.MultiLine** .

24 ToString

25

1  
2  
3 *Description*

4 Occurs when the content of the text box is changed upon server postback.

5 AddAttributesToRender

6  
7 [C#] protected override void AddAttributesToRender(HtmlTextWriter writer);

8 [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);

9 [VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As  
10 HtmlTextWriter)

11 [JScript] protected override function AddAttributesToRender(writer :  
12 HtmlTextWriter);

13  
14 *Description*

15 AddParsedSubObject

16  
17 [C#] protected override void AddParsedSubObject(object obj);

18 [C++] protected: void AddParsedSubObject(Object\* obj);

19 [VB] Overrides Protected Sub AddParsedSubObject(ByVal obj As Object)

20 [JScript] protected override function AddParsedSubObject(obj : Object);

21  
22 *Description*

23 Overridden to only allow literal controls to be added as Text property.

24 OnPreRender

1  
2 [C#] protected override void OnPreRender(EventArgs e);  
3 [C++] protected: void OnPreRender(EventArgs\* e);  
4 [VB] Overrides Protected Sub OnPreRender(ByVal e As EventArgs)  
5 [JScript] protected override function OnPreRender(e : EventArgs);  
6

7 *Description*

8  
9 OnTextChanged

10  
11 [C#] protected virtual void OnTextChanged(EventArgs e);  
12 [C++] protected: virtual void OnTextChanged(EventArgs\* e);  
13 [VB] Overridable Protected Sub OnTextChanged(ByVal e As EventArgs)  
14 [JScript] protected function OnTextChanged(e : EventArgs);  
15

16 *Description*

17 Raises the **TextChanged** event.

18 Render

19  
20 [C#] protected override void Render(HtmlTextWriter writer);  
21 [C++] protected: void Render(HtmlTextWriter\* writer);  
22 [VB] Overrides Protected Sub Render(ByVal writer As HtmlTextWriter)  
23 [JScript] protected override function Render(writer : HtmlTextWriter);  
24

25 *Description*

1 IPostBackDataHandler.LoadPostData

2

3 [C#] bool IPostBackDataHandler.LoadPostData(string postDataKey,  
4 NameValueCollection postCollection);

5 [C++] bool IPostBackDataHandler::LoadPostData(String\* postDataKey,  
6 NameValueCollection\* postCollection);

7 [VB] Function LoadPostData(ByVal postDataKey As String, ByVal  
8 postCollection As NameValueCollection) As Boolean Implements  
9 IPostBackDataHandler.LoadPostData

10 [JScript] function IPostBackDataHandler.LoadPostData(postDataKey : String,  
11 postCollection : NameValueCollection) : Boolean;

12 IPostBackDataHandler.RaisePostDataChangedEvent

13

14 [C#] void IPostBackDataHandler.RaisePostDataChangedEvent();

15 [C++] void IPostBackDataHandler::RaisePostDataChangedEvent();

16 [VB] Sub RaisePostDataChangedEvent() Implements  
17 IPostBackDataHandler.RaisePostDataChangedEvent

18 [JScript] function IPostBackDataHandler.RaisePostDataChangedEvent();

19 TextBoxControlBuilder class (System.Web.UI.WebControls)

20 TrackViewState

21

22

## 23 *Description*

24 Interacts with the parser to build a **System.Web.UI.WebControls.TextBox**  
25 control.



1 To create a custom control builder for a  
2 **System.Web.UI.WebControls.TextBox** derived control, you need to inherit from  
3 this class.

4 **TextBoxControlBuilder**

5 *Example Syntax:*

6 **TrackViewState**

7  
8 [C#] public TextBoxControlBuilder();

9 [C++] public: TextBoxControlBuilder();

10 [VB] Public Sub New()

11 [JScript] public function TextBoxControlBuilder();

12 **ControlType**

13 **FChildrenAsProperties**

14 **FlsNonParserAccessor**

15 **HasAspCode**

16 **ID**

17 **InDesigner**

18 **NamingContainerType**

19 **Parser**

20 **TagName**

21 **AllowWhitespaceLiterals**

22  
23 [C#] public override bool AllowWhitespaceLiterals();

24 [C++] public: bool AllowWhitespaceLiterals();

25 [VB] Overrides Public Function AllowWhitespaceLiterals() As Boolean

1 [JScript] public override function AllowWhitespaceLiterals() : Boolean;

2  
3 *Description*

4 Specifies whether white space literals are allowed.

5 *Return Value:* **false** for all cases.

6 This method overrides

7 **System.Web.UI.ControlBuilder.AllowWhitespaceLiterals** to ignore white  
8 space in the **System.Web.UI.WebControls.TextBox** control.

9 HtmlDecodeLiterals

10  
11 [C#] public override bool HtmlDecodeLiterals();

12 [C++] public: bool HtmlDecodeLiterals();

13 [VB] Overrides Public Function HtmlDecodeLiterals() As Boolean

14 [JScript] public override function HtmlDecodeLiterals() : Boolean;

15  
16 *Description*

17  
18 TextBoxMode enumeration (System.Web.UI.WebControls)

19 ToString

20  
21  
22 *Description*

23 Specifies the behavior mode of the text box.

24 The **System.Web.UI.WebControls.TextBoxMode** enumeration represents  
25 the different display options for **System.Web.UI.WebControls.TextBox** controls.

1 ToString

2

3 [C#] public const TextBoxMode MultiLine;

4 [C++] public: const TextBoxMode MultiLine;

5 [VB] Public Const MultiLine As TextBoxMode

6 [JScript] public var MultiLine : TextBoxMode;

7

8 *Description*

9 Multi line entry mode.

10 ToString

11

12 [C#] public const TextBoxMode Password;

13 [C++] public: const TextBoxMode Password;

14 [VB] Public Const Password As TextBoxMode

15 [JScript] public var Password : TextBoxMode;

16

17 *Description*

18 Password entry mode.

19 ToString

20

21 [C#] public const TextBoxMode SingleLine;

22 [C++] public: const TextBoxMode SingleLine;

23 [VB] Public Const SingleLine As TextBoxMode

24 [JScript] public var SingleLine : TextBoxMode;

25

1  
2 *Description*

3 Single-line entry mode.

4 TitleFormat enumeration (System.Web.UI.WebControls)

5 ToString

6  
7  
8 *Description*

9 Specifies the title format for the displayed month in the  
10 **System.Web.UI.WebControls.Calendar** control.

11 The **System.Web.UI.WebControls.TitleFormat** enumeration represents  
12 the different title formats for the **System.Web.UI.WebControls.Calendar**  
13 control.

14 ToString

15  
16 [C#] public const TitleFormat Month;

17 [C++] public: const TitleFormat Month;

18 [VB] Public Const Month As TitleFormat

19 [JScript] public var Month : TitleFormat;

20  
21 *Description*

22 Title displayed with only the month but not the year. For example,  
23 "January".

24 ToString

```

1
2 [C#] public const TitleFormat MonthYear;
3 [C++] public: const TitleFormat MonthYear;
4 [VB] Public Const MonthYear As TitleFormat
5 [JScript] public var MonthYear : TitleFormat;
6

```

#### *Description*

Title displayed with both the month and the year. For example, "January 2000".

Unit structure (System.Web.UI.WebControls)

ToString

#### *Description*

Defines the fields, properties, and methods of the **System.Web.UI.WebControls.Unit** structure.

ToString

```

18
19 [C#] public static readonly Unit Empty;
20 [C++] public: static Unit Empty;
21 [VB] Public Shared ReadOnly Empty As Unit
22 [JScript] public static var Empty : Unit;
23

```

#### *Description*

Specifies an empty unit. This field is read-only.

Unit

*Example Syntax:*

ToString

[C#] public Unit(double value);

[C++] public: Unit(double value);

[VB] Public Sub New(ByVal value As Double)

[JScript] public function Unit(value : double);

*Description*

Initializes a new instance of the **System.Web.UI.WebControls.Unit** structure with the specified double-precision floating point number as the unit value and **Pixel** as the (default) unit type. Represents the specified unit value.

Unit

*Example Syntax:*

ToString

[C#] public Unit(int value);

[C++] public: Unit(int value);

[VB] Public Sub New(ByVal value As Integer)

[JScript] public function Unit(value : int); Initializes a new instance of the

**System.Web.UI.WebControls.Unit** structure.

*Description*

1        Initializes a new instance of the **System.Web.UI.WebControls.Unit**  
2 structure with the specified 32-bit signed integer as the unit value and **Pixel** as the  
3 (default) unit type. Represents the specified unit value.

4        Unit

5        *Example Syntax:*

6        ToString

7  
8 [C#] public Unit(string value);

9 [C++] public: Unit(String\* value);

10 [VB] Public Sub New(ByVal value As String)

11 [JScript] public function Unit(value : String);

12  
13 *Description*

14        Initializes a new instance of the **System.Web.UI.WebControls.Unit**  
15 structure with the specified text string that contains the unit value and unit type. If  
16 the unit type is not specified, the default is **Pixel** . The specified text string that  
17 contains the unit value and unit type.

18        Unit

19        *Example Syntax:*

20        ToString

21  
22 [C#] public Unit(double value, UnitType type);

23 [C++] public: Unit(double value, UnitType type);

24 [VB] Public Sub New(ByVal value As Double, ByVal type As UnitType)

25 [JScript] public function Unit(value : double, type : UnitType);

1  
2 *Description*

3        Initializes a new instance of the **System.Web.UI.WebControls.Unit**  
4 structure with the specified double-precision floating point number as the unit  
5 value and the specified **System.Web.UI.WebControls.UnitType** as the unit type.  
6 Represents the specified unit value. Represents the specified unit type.

7        Unit

8        *Example Syntax:*

9        ToString

10  
11 [C#] public Unit(string value, CultureInfo culture);

12 [C++] public: Unit(String\* value, CultureInfo\* culture);

13 [VB] Public Sub New(ByVal value As String, ByVal culture As CultureInfo)

14 [JScript] public function Unit(value : String, culture : CultureInfo);

15  
16 *Description*

17  
18        IsEmpty

19        ToString

20  
21 [C#] public bool IsEmpty {get;}

22 [C++] public: \_\_property bool get\_IsEmpty();

23 [VB] Public ReadOnly Property IsEmpty As Boolean

24 [JScript] public function get IsEmpty() : Boolean;



1  
2 *Description*

3 Gets a value indicating whether the **System.Web.UI.WebControls.Unit** is  
4 empty.

5 Type

6 ToString

7  
8 [C#] public UnitType Type {get;}

9 [C++] public: \_\_property UnitType get\_Type();

10 [VB] Public ReadOnly Property Type As UnitType

11 [JScript] public function get Type() : UnitType;

12  
13 *Description*

14 Gets or sets the type of the **System.Web.UI.WebControls.Unit** .

15 Value

16 ToString

17  
18 [C#] public double Value {get;}

19 [C++] public: \_\_property double get\_Value();

20 [VB] Public ReadOnly Property Value As Double

21 [JScript] public function get Value() : double;

22  
23 *Description*

24 Gets the value of the **System.Web.UI.WebControls.Unit** .

25 Equals

1  
2 [C#] public override bool Equals(object obj);

3 [C++] public: bool Equals(Object\* obj);

4 [VB] Overrides Public Function Equals(ByVal obj As Object) As Boolean

5 [JScript] public override function Equals(obj : Object) : Boolean;

6  
7 *Description*

8 Compares this **System.Web.UI.WebControls.Unit** with the specified  
9 object.

10 *Return Value:* **true** if this **System.Web.UI.WebControls.Unit** has the same value  
11 and type as the specified object; otherwise, **false** . The specified object for  
12 comparison.

13 GetHashCode

14  
15 [C#] public override int GetHashCode();

16 [C++] public: int GetHashCode();

17 [VB] Overrides Public Function GetHashCode() As Integer

18 [JScript] public override function GetHashCode() : int;

19  
20 *Description*

21  
22 op\_Equality

23  
24 [C#] public static bool operator ==(Unit left, Unit right);

25 [C++] public: static bool op\_Equality(Unit left, Unit right);

1 [VB] returnValue = Unit.op\_Equality(left, right)

2 [JScript] returnValue = left == right;

3  
4 *Description*

5 Compares two units to find out if they have the same value and type.

6 *Return Value:* **true** if both units have the same value and type; otherwise, **false** .

7 One of the two units being compared. The other of the two units being compared.

8 op\_Implicit

9  
10 [C#] public static implicit operator Unit(int n);

11 [C++] public: static Unit op\_Implicit(int n);

12 [VB] returnValue = Unit.op\_Implicit(n)

13 [JScript] returnValue = n;

14  
15 *Description*

16 Implicitly creates a **System.Web.UI.WebControls.Unit** of type **Pixel** from  
17 the specified 32-bit unsigned integer. The specified 32-bit unsigned integer for  
18 creating a Unit.

19 op\_Inequality

20  
21 [C#] public static bool operator !=(Unit left, Unit right);

22 [C++] public: static bool op\_Inequality(Unit left, Unit right);

23 [VB] returnValue = Unit.op\_Inequality(left, right)

24 [JScript] returnValue = left != right;

1  
2 *Description*

3 Compares two units to find out if they have different values and/or types.  
4 *Return Value:* **true** if both units have different values or different types; otherwise,  
5 **false** . One of the two units being compared. The other of the two units being  
6 compared.

7 Parse

8  
9 [C#] public static Unit Parse(string s);  
10 [C++] public: static Unit Parse(String\* s);  
11 [VB] Public Shared Function Parse(ByVal s As String) As Unit  
12 [JScript] public static function Parse(s : String) : Unit;  
13

14 *Description*

15  
16 Parse

17  
18 [C#] public static Unit Parse(string s, CultureInfo culture);  
19 [C++] public: static Unit Parse(String\* s, CultureInfo\* culture);  
20 [VB] Public Shared Function Parse(ByVal s As String, ByVal culture As  
21 CultureInfo) As Unit  
22 [JScript] public static function Parse(s : String, culture : CultureInfo) : Unit;  
23

24 *Description*  
25

## Percentage

```
[C#] public static Unit Percentage(double n);  
[C++] public: static Unit Percentage(double n);  
[VB] Public Shared Function Percentage(ByVal n As Double) As Unit  
[JScript] public static function Percentage(n : double) : Unit;
```

### *Description*

Creates a **System.Web.UI.WebControls.Unit** of type **Percentage** from the specified 32-bit signed integer.

*Return Value:* A **System.Web.UI.WebControls.Unit** that represents the unit created. The specified double precision floating point number for creating a unit.

## Pixel

```
[C#] public static Unit Pixel(int n);  
[C++] public: static Unit Pixel(int n);  
[VB] Public Shared Function Pixel(ByVal n As Integer) As Unit  
[JScript] public static function Pixel(n : int) : Unit;
```

### *Description*

Creates a **System.Web.UI.WebControls.Unit** of type **Pixel** from the specified 32-bit signed integer.

*Return Value:* A **System.Web.UI.WebControls.Unit** that represents the unit created. The specified 32-bit signed integer for creating a unit.

## Point

1  
2 [C#] public static Unit Point(int n);

3 [C++] public: static Unit Point(int n);

4 [VB] Public Shared Function Point(ByVal n As Integer) As Unit

5 [JScript] public static function Point(n : int) : Unit;

6  
7 *Description*

8       Creates a **System.Web.UI.WebControls.Unit** of type **Point** from the  
9 specified 32-bit signed integer.

10 *Return Value:* A **System.Web.UI.WebControls.Unit** that represents the unit  
11 created. The specified 32-bit signed integer for creating a unit.

12       ToString

13  
14 [C#] public override string ToString();

15 [C++] public: String\* ToString();

16 [VB] Overrides Public Function ToString() As String

17 [JScript] public override function ToString() : String;

18  
19 *Description*

20       Converts a **System.Web.UI.WebControls.Unit** to a **System.String** .

21 *Return Value:* A **System.String** represents this

22 **System.Web.UI.WebControls.Unit** .

23       ToString

24  
25 [C#] public string ToString(CultureInfo culture);

1 [C++] public: String\* ToString(CultureInfo\* culture);

2 [VB] Public Function ToString(ByVal culture As CultureInfo) As String

3 [JScript] public function ToString(culture : CultureInfo) : String;

4  
5 *Description*

6  
7 UnitConverter class (System.Web.UI.WebControls)

8 ToString

9  
10  
11 *Description*

12 Specifies an interface to be overridden to provide unit conversion services.

13 The base unit converter class.

14 UnitConverter

15 *Example Syntax:*

16 ToString

17  
18 [C#] public UnitConverter();

19 [C++] public: UnitConverter();

20 [VB] Public Sub New()

21 [JScript] public function UnitConverter();

22 CanConvertFrom

23  
24 [C#] public override bool CanConvertFrom(ITypeDescriptorContext context,

25 Type sourceType);

1 [C++] public: bool CanConvertFrom(ITypeDescriptorContext\* context, Type\*  
2 sourceType);

3 [VB] Overrides Public Function CanConvertFrom(ByVal context As  
4 ITypeDescriptorContext, ByVal sourceType As Type) As Boolean

5 [JScript] public override function CanConvertFrom(context :  
6 ITypeDescriptorContext, sourceType : Type) : Boolean;

### 8 *Description*

9 Returns a value indicating whether the unit converter can convert from the  
10 specified source type.

11 *Return Value:* **true** if the source type can be converted from; otherwise, **false** . An  
12 **System.ComponentModel.ITypeDescriptorContext** that specifies the context of  
13 the object to convert. The type of the source.

### 14 *ConvertFrom*

15  
16 [C#] public override object ConvertFrom(ITypeDescriptorContext context,  
17 CultureInfo culture, object value);

18 [C++] public: Object\* ConvertFrom(ITypeDescriptorContext\* context,  
19 CultureInfo\* culture, Object\* value);

20 [VB] Overrides Public Function ConvertFrom(ByVal context As  
21 ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object)  
22 As Object

23 [JScript] public override function ConvertFrom(context : ITypeDescriptorContext,  
24 culture : CultureInfo, value : Object) : Object;



## Description

Performs type conversion from the specified context, object and argument list.

**Return Value:** The object resulting from conversion. An **System.ComponentModel.ITypeDescriptorContext** that indicates the context of the object to convert. A **System.Globalization.CultureInfo** object that represents information about a culture such as language, calendar system, and so on. This parameter is not used in this method. It is reserved for future versions of this method. You can optionally pass in **null** for this parameter. The object to convert.

## ConvertTo

[C#] public override object ConvertTo(ITypeDescriptorContext context, CultureInfo culture, object value, Type destinationType);

[C++] public: Object\* ConvertTo(ITypeDescriptorContext\* context, CultureInfo\* culture, Object\* value, Type\* destinationType);

[VB] Overrides Public Function ConvertTo(ByVal context As ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object, ByVal destinationType As Type) As Object

[JScript] public override function ConvertTo(context : ITypeDescriptorContext, culture : CultureInfo, value : Object, destinationType : Type) : Object;

## Description

Performs type conversion to the specified destination type given the specified context, object and argument list.

*Return Value:* The object resulting from conversion. An **System.ComponentModel.ITypeDescriptorContext** that indicates the context of the object to convert. A **System.Globalization.CultureInfo** object that represents information about a culture such as language, calendar system, and so on. This parameter is not used in this method. It is reserved for future versions of this method. You can optionally pass in **null** for this parameter. The object to convert. The type to convert to.

UnitType enumeration (System.Web.UI.WebControls)

ToString

#### *Description*

Specifies the unit of measurement.

The **System.Web.UI.WebControls.UnitType** enumeration represents the different supported measurement units. Measurements can be represented in pixels, points, picas, inches, millimeters, centimeters, percentages, em, or ex.

ToString

[C#] public const UnitType Cm;

[C++] public: const UnitType Cm;

[VB] Public Const Cm As UnitType

[JScript] public var Cm : UnitType;

#### *Description*

Measurement is in centimeters.

ToString

[C#] public const UnitType Em;  
[C++] public: const UnitType Em;  
[VB] Public Const Em As UnitType  
[JScript] public var Em : UnitType;

*Description*

Measurement is relative to the height of the parent element's font.

ToString

[C#] public const UnitType Ex;  
[C++] public: const UnitType Ex;  
[VB] Public Const Ex As UnitType  
[JScript] public var Ex : UnitType;

*Description*

Measurement is relative to the height of the lowercase letter x of the parent element's font.

ToString

[C#] public const UnitType Inch;  
[C++] public: const UnitType Inch;  
[VB] Public Const Inch As UnitType  
[JScript] public var Inch : UnitType;

1  
2 *Description*

3 Measurement is in inches.

4 ToString

5  
6 [C#] public const UnitType Mm;

7 [C++] public: const UnitType Mm;

8 [VB] Public Const Mm As UnitType

9 [JScript] public var Mm : UnitType;

10  
11 *Description*

12 Measurement is in millimeters.

13 ToString

14  
15 [C#] public const UnitType Percentage;

16 [C++] public: const UnitType Percentage;

17 [VB] Public Const Percentage As UnitType

18 [JScript] public var Percentage : UnitType;

19  
20 *Description*

21 Measurement is a percentage relative to the parent element.

22 ToString

23  
24 [C#] public const UnitType Pica;

25 [C++] public: const UnitType Pica;

1 [VB] Public Const Pica As UnitType

2 [JScript] public var Pica : UnitType;

4 *Description*

5 Measurement is in picas. A pica represents 12 points.

6 ToString

8 [C#] public const UnitType Pixel;

9 [C++] public: const UnitType Pixel;

10 [VB] Public Const Pixel As UnitType

11 [JScript] public var Pixel : UnitType;

13 *Description*

14 Measurement is in pixels.

15 ToString

17 [C#] public const UnitType Point;

18 [C++] public: const UnitType Point;

19 [VB] Public Const Point As UnitType

20 [JScript] public var Point : UnitType;

22 *Description*

23 Measurement is in points. A point represents 1/72 of an inch.

24 ValidatedControlConverter class (System.Web.UI.WebControls)

25 ToString

1  
2  
3 *Description*

4 Shows a list of validatable controls in the Properties window.

5 GetStandardValues

6  
7 [C#] public override StandardValueCollection

8 GetStandardValues(ITypeDescriptorContext context);

9 [C++] public: StandardValueCollection\*

10 GetStandardValues(ITypeDescriptorContext\* context);

11 [VB] Overrides Public Function GetStandardValues(ByVal context As

12 ITypeDescriptorContext) As StandardValueCollection

13 [JScript] public override function GetStandardValues(context :

14 ITypeDescriptorContext) : StandardValueCollection;

15  
16 *Description*

17 Returns a collection of standard values retrieved from the context specified  
18 by the specified type descriptor.

19 *Return Value:* A StandardValueCollection that represents the standard values  
20 collected from the specified context. A type descriptor that specifies the location  
21 of the context to convert from.

22 GetStandardValuesExclusive

23  
24 [C#] public override bool GetStandardValuesExclusive(ITypeDescriptorContext  
25 context);

```

1 [C++] public: bool GetStandardValuesExclusive(ITypeDescriptorContext*
2 context);
3 [VB] Overrides Public Function GetStandardValuesExclusive(ByVal context As
4 ITypeDescriptorContext) As Boolean
5 [JScript] public override function GetStandardValuesExclusive(context :
6 ITypeDescriptorContext) : Boolean;
7

```

#### *Description*

Gets whether or not the context specified contains exclusive standard values.

**Return Value:** **true** if the specified context contains exclusive standard values, otherwise **false** . A type descriptor that indicates the context to convert from.

#### **GetStandardValuesSupported**

```

15 [C#] public override bool GetStandardValuesSupported(ITypeDescriptorContext
16 context);
17 [C++] public: bool GetStandardValuesSupported(ITypeDescriptorContext*
18 context);
19 [VB] Overrides Public Function GetStandardValuesSupported(ByVal context As
20 ITypeDescriptorContext) As Boolean
21 [JScript] public override function GetStandardValuesSupported(context :
22 ITypeDescriptorContext) : Boolean;
23

```

#### *Description*

Gets whether or not the specified context contains supported standard values.

*Return Value:* **true** if the specified context contains supported standard values, otherwise **false**. A type descriptor that indicates the context to convert from.

ValidationCompareOperator enumeration (System.Web.UI.WebControls)

ToString

#### *Description*

Specifies the validation comparison operators used by the **System.Web.UI.WebControls.CompareValidator** control.

The **System.Web.UI.WebControls.ValidationCompareOperator** enumeration represents the comparison operations that can be performed by the **System.Web.UI.WebControls.CompareValidator** control.

ToString

[C#] public const ValidationCompareOperator DataTypeCheck;

[C++] public: const ValidationCompareOperator DataTypeCheck;

[VB] Public Const DataTypeCheck As ValidationCompareOperator

[JScript] public var DataTypeCheck : ValidationCompareOperator;

#### *Description*

A comparison for data type only.

ToString



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

```
[C#] public const ValidationCompareOperator Equal;  
[C++] public: const ValidationCompareOperator Equal;  
[VB] Public Const Equal As ValidationCompareOperator  
[JScript] public var Equal : ValidationCompareOperator;
```

*Description*

A comparison for equality.  
ToString

```
[C#] public const ValidationCompareOperator GreaterThan;  
[C++] public: const ValidationCompareOperator GreaterThan;  
[VB] Public Const GreaterThan As ValidationCompareOperator  
[JScript] public var GreaterThan : ValidationCompareOperator;
```

*Description*

A comparison for greater than.  
ToString

```
[C#] public const ValidationCompareOperator GreaterThanEqual;  
[C++] public: const ValidationCompareOperator GreaterThanEqual;  
[VB] Public Const GreaterThanEqual As ValidationCompareOperator  
[JScript] public var GreaterThanEqual : ValidationCompareOperator;
```

*Description*

A comparison for greater than or equal to.

ToString

[C#] public const ValidationCompareOperator LessThan;

[C++] public: const ValidationCompareOperator LessThan;

[VB] Public Const LessThan As ValidationCompareOperator

[JScript] public var LessThan : ValidationCompareOperator;

*Description*

A comparison for less than.

ToString

[C#] public const ValidationCompareOperator LessThanEqual;

[C++] public: const ValidationCompareOperator LessThanEqual;

[VB] Public Const LessThanEqual As ValidationCompareOperator

[JScript] public var LessThanEqual : ValidationCompareOperator;

*Description*

A comparison for less than or equal to.

ToString

[C#] public const ValidationCompareOperator NotEqual;

[C++] public: const ValidationCompareOperator NotEqual;

[VB] Public Const NotEqual As ValidationCompareOperator

[JScript] public var NotEqual : ValidationCompareOperator;

1  
2 *Description*

3 A comparison for inequality.

4 ValidationDataType enumeration (System.Web.UI.WebControls)

5 ToString

6  
7  
8 *Description*

9 Specifies the validation data types used by the

10 **System.Web.UI.WebControls.CompareValidator** and

11 **System.Web.UI.WebControls.RangeValidator** controls.

12 The **System.Web.UI.WebControls.ValidationDataType** enumeration  
13 represents the different data types that the

14 **System.Web.UI.WebControls.CompareValidator** and

15 **System.Web.UI.WebControls.RangeValidator** controls can validate.

16 ToString

17  
18 [C#] public const ValidationDataType Currency;

19 [C++] public: const ValidationDataType Currency;

20 [VB] Public Const Currency As ValidationDataType

21 [JScript] public var Currency : ValidationDataType;

22  
23 *Description*

24 A currency data type.

25 ToString

1  
2 [C#] public const ValidationDataType Date;  
3 [C++] public: const ValidationDataType Date;  
4 [VB] Public Const Date As ValidationDataType  
5 [JScript] public var Date : ValidationDataType;

6  
7 *Description*

8 A date data type.  
9 ToString

10  
11 [C#] public const ValidationDataType Double;  
12 [C++] public: const ValidationDataType Double;  
13 [VB] Public Const Double As ValidationDataType  
14 [JScript] public var Double : ValidationDataType;

15  
16 *Description*

17 A double data type.  
18 ToString

19  
20 [C#] public const ValidationDataType Integer;  
21 [C++] public: const ValidationDataType Integer;  
22 [VB] Public Const Integer As ValidationDataType  
23 [JScript] public var Integer : ValidationDataType;

24  
25 *Description*

An integer data type.

ToString

[C#] public const ValidationDataType String;

[C++] public: const ValidationDataType String;

[VB] Public Const String As ValidationDataType

[JScript] public var String : ValidationDataType;

#### *Description*

A string data type.

ValidationSummary class (System.Web.UI.WebControls)

ToString

#### *Description*

Displays a summary of all validation errors inline on a Web page, in a message box, or both.

The **System.Web.UI.WebControls.ValidationSummary** class is used to summarize the error messages from all validators on a Web page, in a single location. The summary can be displayed as a list, as a bulleted list, or as a single paragraph based on the

**System.Web.UI.WebControls.ValidationSummary.DisplayMode** property.

ValidationSummary

*Example Syntax:*

ToString

```

1
2 [C#] public ValidationSummary();
3 [C++] public: ValidationSummary();
4 [VB] Public Sub New()
5 [JScript] public function ValidationSummary();
6

```

### *Description*

Initializes a new instance of the **System.Web.UI.WebControls.ValidationSummary** class.

The following table shows the initial property value for an instance of **System.Web.UI.WebControls.ValidationSummary**.

AccessKey
Attributes
BackColor
BorderColor
BorderStyle
BorderWidth
ChildControlsCreated
ClientID
Context
Controls
ControlStyle
ControlStyleCreated
CssClass
DisplayMode

ToString

*Description*

Gets or sets the display mode of the validation summary.

Use this property to specify the display format of a

**System.Web.UI.WebControls.ValidationSummary** control. The summary may be displayed as a list, as a bulleted list, or as a single paragraph.

EnableClientScript

ToString

[C#] public bool EnableClientScript {get; set;}

[C++] public: \_\_property bool get\_EnableClientScript();public: \_\_property void set\_EnableClientScript(bool);

[VB] Public Property EnableClientScript As Boolean

[JScript] public function get EnableClientScript() : Boolean;public function set EnableClientScript(Boolean);

*Description*

Gets or sets a value indicating whether client-side validation is attempted on the browser.

Use this property to enable or disable client-side validation on the browser.

When set to **true** , client-side validation is performed on the browser if the browser supports that feature. When set to **false** , no client-side validation is attempted, the **System.Web.UI.WebControls.ValidationSummary** control only

updates itself on round-trips to the server, and the **System.Web.UI.WebControls.ValidationSummary.ShowMessageBox** property has no effect.

Enabled

EnableViewState

Events

Font

ForeColor

ToString

#### *Description*

Gets or sets the fore color of the control.

Use this property to specify the color that error messages from validation controls are displayed.

HasChildViewState

HeaderText

ToString

#### *Description*

Gets or sets the header text displayed at the top of the summary.

Use this property to display a title for the

**System.Web.UI.WebControls.ValidationSummary** control.

Height



1 ID  
2 IsTrackingViewState  
3 NamingContainer  
4 Page  
5 Parent  
6 ShowMessageBox  
7 ToString

8  
9  
10 *Description*

11 Gets or sets a value indicating whether the validation summary is displayed  
12 in a message box.

13 This property can be used in addition to the  
14 **System.Web.UI.WebControls.ValidationSummary.ShowSummary** property to  
15 control where the validation summary is displayed. If this property and  
16 **System.Web.UI.WebControls.ValidationSummary.EnableClientScript** are  
17 both set to **true**, the validation summary is displayed in a message box. If  
18 **System.Web.UI.WebControls.ValidationSummary.EnableClientScript** is set  
19 to **false** , this property has no effect.

20 ShowSummary

21 ToString

22  
23 [C#] public bool ShowSummary {get; set;}

24 [C++] public: \_\_property bool get\_ShowSummary();public: \_\_property void

25 set\_ShowSummary(bool);

1 [VB] Public Property ShowSummary As Boolean

2 [JScript] public function get ShowSummary() : Boolean;public function set

3 ShowSummary(Boolean);

4  
5 *Description*

6 Gets or sets a value indicating whether the validation summary is displayed  
7 inline.

8 This property can be used in addition to the  
9 **System.Web.UI.WebControls.ValidationSummary.ShowMessageBox** property  
10 to control where the validation summary is displayed. If this property is set to **true**  
11 , the validation summary is displayed on the Web page.

12 Site

13 Style

14 TabIndex

15 TagKey

16 TagName

17 TemplateSourceDirectory

18 ToolTip

19 UniqueID

20 ViewState

21 ViewStateIgnoresCase

22 Visible

23 Width

24 AddAttributesToRender

25

1  
2 [C#] protected override void AddAttributesToRender(HtmlTextWriter writer);

3 [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);

4 [VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As  
5 HtmlTextWriter)

6 [JScript] protected override function AddAttributesToRender(writer :  
7 HtmlTextWriter);

8  
9 *Description*

10 AddAttributesToRender method.

11 OnPreRender

12  
13 [C#] protected override void OnPreRender(EventArgs e);

14 [C++] protected: void OnPreRender(EventArgs\* e);

15 [VB] Overrides Protected Sub OnPreRender(ByVal e As EventArgs)

16 [JScript] protected override function OnPreRender(e : EventArgs);

17  
18 *Description*

19 PreRender method.

20 Render

21  
22 [C#] protected override void Render(HtmlTextWriter writer);

23 [C++] protected: void Render(HtmlTextWriter\* writer);

24 [VB] Overrides Protected Sub Render(ByVal writer As HtmlTextWriter)

25 [JScript] protected override function Render(writer : HtmlTextWriter);

1  
2 *Description*

3       Render method.

4       ValidationSummaryDisplayMode enumeration  
5 (System.Web.UI.WebControls)

6       TrackViewState  
7  
8

9 *Description*

10       Specifies the validation summary display mode used by the  
11 **System.Web.UI.WebControls.ValidationSummary** control.

12       The **System.Web.UI.WebControls.ValidationSummaryDisplayMode**  
13 enumeration represents the different display formats of a  
14 **System.Web.UI.WebControls.ValidationSummary** control.

15       TrackViewState  
16

17 [C#] public const ValidationSummaryDisplayMode BulletList;

18 [C++] public: const ValidationSummaryDisplayMode BulletList;

19 [VB] Public Const BulletList As ValidationSummaryDisplayMode

20 [JScript] public var BulletList : ValidationSummaryDisplayMode;

21  
22 *Description*

23       Validation summary displayed in a bulleted list.

24       TrackViewState  
25

1  
2 [C#] public const ValidationSummaryDisplayMode List;  
3 [C++] public: const ValidationSummaryDisplayMode List;  
4 [VB] Public Const List As ValidationSummaryDisplayMode  
5 [JScript] public var List : ValidationSummaryDisplayMode;

6  
7 *Description*

8 Validation summary displayed in a list.

9 TrackViewState

10  
11 [C#] public const ValidationSummaryDisplayMode SingleParagraph;  
12 [C++] public: const ValidationSummaryDisplayMode SingleParagraph;  
13 [VB] Public Const SingleParagraph As ValidationSummaryDisplayMode  
14 [JScript] public var SingleParagraph : ValidationSummaryDisplayMode;

15  
16 *Description*

17 Validation summary displayed in a single paragraph.

18 ValidatorDisplay enumeration (System.Web.UI.WebControls)

19 ToString

20  
21  
22 *Description*

23 Specifies the display behavior of error messages in validation controls.

24 The **ValidatorDisplaySystem.Web.UI.WebControls** enumeration  
25 represents the different display behaviors of error messages in validation controls.

1 ToString

2

3 [C#] public const ValidatorDisplay Dynamic;

4 [C++] public: const ValidatorDisplay Dynamic;

5 [VB] Public Const Dynamic As ValidatorDisplay

6 [JScript] public var Dynamic : ValidatorDisplay;

7

8 *Description*

9 Validator content dynamically added to the page when validation fails.

10 ToString

11

12 [C#] public const ValidatorDisplay None;

13 [C++] public: const ValidatorDisplay None;

14 [VB] Public Const None As ValidatorDisplay

15 [JScript] public var None : ValidatorDisplay;

16

17 *Description*

18 Validator content never displayed inline.

19 ToString

20

21 [C#] public const ValidatorDisplay Static;

22 [C++] public: const ValidatorDisplay Static;

23 [VB] Public Const Static As ValidatorDisplay

24 [JScript] public var Static : ValidatorDisplay;

25

1  
2 *Description*

3 Validator content physically part of the page layout.

4 VerticalAlign enumeration (System.Web.UI.WebControls)

5 ToString

6  
7  
8 *Description*

9 Specifies the vertical alignment of an object or text in a control.

10 The **System.Web.UI.WebControls.VerticalAlign** enumeration represents  
11 the different vertical alignment options for an object or text in a control.

12 ToString

13  
14 [C#] public const VerticalAlign Bottom;

15 [C++] public: const VerticalAlign Bottom;

16 [VB] Public Const Bottom As VerticalAlign

17 [JScript] public var Bottom : VerticalAlign;

18  
19 *Description*

20 Text or object is aligned with the bottom of the enclosing control.

21 ToString

22  
23 [C#] public const VerticalAlign Middle;

24 [C++] public: const VerticalAlign Middle;

25 [VB] Public Const Middle As VerticalAlign

1 [JScript] public var Middle : VerticalAlign;

3 *Description*

4 Text or object is aligned with the center of the enclosing control.

5 ToString

7 [C#] public const VerticalAlign NotSet;

8 [C++] public: const VerticalAlign NotSet;

9 [VB] Public Const NotSet As VerticalAlign

10 [JScript] public var NotSet : VerticalAlign;

12 *Description*

13 Vertical alignment is not set.

14 ToString

16 [C#] public const VerticalAlign Top;

17 [C++] public: const VerticalAlign Top;

18 [VB] Public Const Top As VerticalAlign

19 [JScript] public var Top : VerticalAlign;

21 *Description*

22 Text or object is aligned with the top of the enclosing control.

23 WebColorConverter class (System.Web.UI.WebControls)

24 ToString



1  
2  
3 *Description*

4 WebColorConverter

5 *Example Syntax:*

6 ToString

7  
8 [C#] public WebColorConverter();

9 [C++] public: WebColorConverter();

10 [VB] Public Sub New()

11 [JScript] public function WebColorConverter();

12 ConvertFrom

13  
14 [C#] public override object ConvertFrom(ITypeDescriptorContext context,  
15 CultureInfo culture, object value);

16 [C++] public: Object\* ConvertFrom(ITypeDescriptorContext\* context,  
17 CultureInfo\* culture, Object\* value);

18 [VB] Overrides Public Function ConvertFrom(ByVal context As  
19 ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object)  
20 As Object

21 [JScript] public override function ConvertFrom(context : ITypeDescriptorContext,  
22 culture : CultureInfo, value : Object) : Object;

23  
24 *Description*

25 ConvertTo

```

1
2 [C#] public override object ConvertTo(ITypeDescriptorContext context,
3   CultureInfo culture, object value, Type destinationType);
4 [C++] public: Object* ConvertTo(ITypeDescriptorContext* context, CultureInfo*
5   culture, Object* value, Type* destinationType);
6 [VB] Overrides Public Function ConvertTo(ByVal context As
7   ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object,
8   ByVal destinationType As Type) As Object
9 [JScript] public override function ConvertTo(context : ITypeDescriptorContext,
10  culture : CultureInfo, value : Object, destinationType : Type) : Object;

```

#### *Description*

WebControl class (System.Web.UI.WebControls)

ToString

#### *Description*

Serves as the base class that defines the methods, properties and events common to all controls in the **System.Web.UI.WebControls** namespace.

Certain properties of the base control may not render on downlevel browsers for some or all controls. For example, the **System.Web.UI.WebControls.WebControl.AccessKey** property will not render on downlevel browsers for any controls. See specific property for more details.

WebControl

*Example Syntax:*

1 ToString

2

3 [C#] protected WebControl();

4 [C++] protected: WebControl();

5 [VB] Protected Sub New()

6 [JScript] protected function WebControl(); Initializes a new instance of the

7 **System.Web.UI.WebControls.WebControl** class.

8

9 *Description*

10 Initializes a new instance of the

11 **System.Web.UI.WebControls.WebControl** class with a **Span** HTML tag.

12 WebControl

13 *Example Syntax:*

14 ToString

15

16 [C#] public WebControl(HtmlTextWriterTag tag);

17 [C++] public: WebControl(HtmlTextWriterTag tag);

18 [VB] Public Sub New(ByVal tag As HtmlTextWriterTag)

19 [JScript] public function WebControl(tag : HtmlTextWriterTag);

20

21 *Description*

22 Initializes a new instance of the

23 **System.Web.UI.WebControls.WebControl** class using the specified HTML tag.

24 One of the **System.Web.UI.HtmlTextWriterTag** values.

25 WebControl

*Example Syntax:*

ToString

[C#] protected WebControl(string tag);

[C++] protected: WebControl(String\* tag);

[VB] Protected Sub New(ByVal tag As String)

[JScript] protected function WebControl(tag : String);

*Description*

Initializes a new instance of the

**System.Web.UI.WebControls.WebControl** class with the specified HTML tag.

An HTML tag.

AccessKey

ToString

[C#] public virtual string AccessKey {get; set;}

[C++] public: \_\_property virtual String\* get\_AccessKey();public: \_\_property

virtual void set\_AccessKey(String\*);

[VB] Overridable Public Property AccessKey As String

[JScript] public function get AccessKey() : String;public function set

AccessKey(String);

*Description*

Gets or sets the keyboard shortcut key (AccessKey) for setting focus to the

Web control.

1 This property will not render on downlevel browsers for any controls. It is  
2 not HTML 4.0 and will only work in IE 4 or higher.

3 Attributes

4 ToString

5  
6 [C#] public AttributeCollection Attributes {get;}

7 [C++] public: \_\_property AttributeCollection\* get\_Attributes();

8 [VB] Public ReadOnly Property Attributes As AttributeCollection

9 [JScript] public function get Attributes() : AttributeCollection;

10  
11 *Description*

12 Gets the collection of arbitrary attributes (for rendering only) that do not  
13 correspond to properties on the control.

14 This property will render on downlevel browsers for all controls.

15 BackColor

16 ToString

17  
18 [C#] public virtual Color BackColor {get; set;}

19 [C++] public: \_\_property virtual Color get\_BackColor();public: \_\_property virtual  
20 void set\_BackColor(Color);

21 [VB] Overridable Public Property BackColor As Color

22 [JScript] public function get BackColor() : Color;public function set  
23 BackColor(Color);

24  
25 *Description*

Gets or sets the background color of the Web control.

This property will render on downlevel browsers for some controls only.

For example, **System.Web.UI.WebControls.Table** ,  
**System.Web.UI.WebControls.Panel** , **System.Web.UI.WebControls.DataGrid**  
, **System.Web.UI.WebControls.Calendar** , and  
**System.Web.UI.WebControls.ValidationSummary** . It will also work for  
**System.Web.UI.WebControls.CheckBoxList** ,  
**System.Web.UI.WebControls.RadioButtonList** and  
**System.Web.UI.WebControls.DataList** if their **RepeatLayout** property is **Table**  
and not **Flow** .

**BorderColor**

**ToString**

[C#] public virtual Color BorderColor {get; set;}

[C++] public: \_\_property virtual Color get\_BorderColor();public: \_\_property

virtual void set\_BorderColor(Color);

[VB] Overridable Public Property BorderColor As Color

[JScript] public function get BorderColor() : Color;public function set

BorderColor(Color);

### *Description*

Gets or sets the border color of the Web control.

The property will render downlevel only for the same table-based controls  
like the **System.Web.UI.WebControls.WebControl.BackColor** property.

However, it is output as the "bordercolor" attribute which is not part of the HTML

3.2 standard. It works for Navigaor 4 and higher, and IE 3 and higher, but not most other browsers.

BorderStyle

ToString

```
[C#] public virtual BorderStyle BorderStyle {get; set;}
```

```
[C++] public: __property virtual BorderStyle get_BorderStyle();public:
```

```
__property virtual void set_BorderStyle(BorderStyle);
```

```
[VB] Overridable Public Property BorderStyle As BorderStyle
```

```
[JScript] public function get BorderStyle() : BorderStyle;public function set
```

```
BorderStyle(BorderStyle);
```

### *Description*

Gets or sets the border style of the Web control.

This property will not render on downlevel browsers for any controls.

BorderWidth

ToString

```
[C#] public virtual Unit BorderWidth {get; set;}
```

```
[C++] public: __property virtual Unit get_BorderWidth();public: __property
```

```
virtual void set_BorderWidth(Unit);
```

```
[VB] Overridable Public Property BorderWidth As Unit
```

```
[JScript] public function get BorderWidth() : Unit;public function set
```

```
BorderWidth(Unit);
```

1  
2 *Description*

3 Gets or sets the border width of the Web control.

4 Use the **System.Web.UI.WebControls.WebControl.BorderWidth**  
5 property to specify a border width for a control.

6 ChildControlsCreated

7 ClientID

8 Context

9 Controls

10 ControlStyle

11 ToString

12  
13  
14 *Description*

15 Gets the style of the Web control. This property is primarily used by control  
16 developers.

17 ControlStyleCreated

18 ToString

19  
20 [C#] public bool ControlStyleCreated {get;}

21 [C++] public: \_\_property bool get\_ControlStyleCreated();

22 [VB] Public ReadOnly Property ControlStyleCreated As Boolean

23 [JScript] public function get ControlStyleCreated() : Boolean;

24  
25 *Description*



CssClass

ToString

[C#] public virtual string CssClass {get; set;}

[C++] public: \_\_property virtual String\* get\_CssClass();public: \_\_property virtual  
void set\_CssClass(String\*);

[VB] Overridable Public Property CssClass As String

[JScript] public function get CssClass() : String;public function set  
CssClass(String);

#### *Description*

Gets or sets the CSS class rendered by the Web control.

This property will render on downlevel browsers for all controls.

Enabled

ToString

[C#] public virtual bool Enabled {get; set;}

[C++] public: \_\_property virtual bool get\_Enabled();public: \_\_property virtual  
void set\_Enabled(bool);

[VB] Overridable Public Property Enabled As Boolean

[JScript] public function get Enabled() : Boolean;public function set  
Enabled(Boolean);

#### *Description*

Gets or sets a value indicating whether the Web control is enabled.

This property will render on downlevel browsers for some controls only.

EnableViewState

Events

Font

ToString

### *Description*

Gets font information of the Web control.

This property includes subproperties that can be accessed declaratively in the form of "Font-Bold" or programmatically in the form of "Font.Bold".

ForeColor

ToString

```
[C#] public virtual Color ForeColor {get; set;}
```

```
[C++] public: __property virtual Color get_ForeColor();public: __property virtual  
void set_ForeColor(Color);
```

```
[VB] Overridable Public Property ForeColor As Color
```

```
[JScript] public function get ForeColor() : Color;public function set  
ForeColor(Color);
```

### *Description*

Gets or sets the foreground color (typically the color of the text) of the Web control.

1 This property will render on downlevel browsers for almost all controls,  
2 except the **System.Web.UI.WebControls.Image** ,  
3 **System.Web.UI.WebControls.AdRotator** ,  
4 **System.Web.UI.WebControls.HyperLink** and  
5 **System.Web.UI.WebControls.LinkButton** controls. In addition, it will be  
6 rendered as tags on downlevel browsers.

7 HasChildViewState

8 Height

9 ToString

10  
11  
12 *Description*

13 Gets or sets the height of the Web control.

14 This property will render on downlevel browsers for some controls only. It  
15 will not render downlevel for **System.Web.UI.WebControls.Label** ,  
16 **System.Web.UI.WebControls.HyperLink** ,  
17 **System.Web.UI.WebControls.LinkButton** , any validator controls, or for  
18 **System.Web.UI.WebControls.CheckBoxList** ,  
19 **System.Web.UI.WebControls.RadioButtonList** and  
20 **System.Web.UI.WebControls.DataList** when their **RepeatLayout** property is  
21 **Flow** . Furthermore, only unit types of **Pixel** and **percentage** will work.

22 ID

23 IsTrackingViewState

24 NamingContainer

25 Page

1 Parent  
2 Site  
3 Style  
4 ToString  
5  
6

7 *Description*

8 Gets a collection of text attributes that will be rendered as a style attribute  
9 on the outer tag of the Web control.

10 This property will render on downlevel browsers for all controls.

11 TabIndex

12 ToString  
13

14 [C#] public virtual short TabIndex {get; set;}

15 [C++] public: \_\_property virtual short get\_TabIndex();public: \_\_property virtual  
16 void set\_TabIndex(short);

17 [VB] Overridable Public Property TabIndex As Short

18 [JScript] public function get TabIndex() : Int16;public function set  
19 TabIndex(Int16);  
20

21 *Description*

22 Gets or sets the tab index of the Web control.

23 Use the **System.Web.UI.WebControls.WebControl.TabIndex** property  
24 to specify or determine the tab index of a control on the Web Page.

25 TagKey

1 ToString

2

3 [C#] protected virtual HtmlTextWriterTag TagKey {get;}

4 [C++] protected: \_\_property virtual HtmlTextWriterTag get\_TagKey();

5 [VB] Overridable Protected ReadOnly Property TagKey As HtmlTextWriterTag

6 [JScript] protected function get TagKey() : HtmlTextWriterTag;

7

8 *Description*

9 TagName

10 ToString

11

12 [C#] protected virtual string TagName {get;}

13 [C++] protected: \_\_property virtual String\* get\_TagName();

14 [VB] Overridable Protected ReadOnly Property TagName As String

15 [JScript] protected function get TagName() : String;

16

17 *Description*

18 A protected property. Gets the name of the control tag. This property is

19 primarily used by control developers.

20 TemplateSourceDirectory

21 ToolTip

22 ToString

23

24

25 *Description*

1 Gets or sets the tool tip for the Web control to be displayed when the mouse  
2 cursor is over the control.

3 This property will not render on downlevel browsers for any controls.

4 UniqueID

5 ViewState

6 ViewStateIgnoresCase

7 Visible

8 Width

9 ToString

10  
11  
12 *Description*

13 Gets or sets the width of the Web control.

14 This property will render on downlevel browsers for some controls only. It  
15 will not render downlevel for **System.Web.UI.WebControls.Label** ,  
16 **System.Web.UI.WebControls.HyperLink** ,  
17 **System.Web.UI.WebControls.LinkButton** , any validator controls, or for  
18 **System.Web.UI.WebControls.CheckBoxList** ,  
19 **System.Web.UI.WebControls.RadioButtonList** and  
20 **System.Web.UI.WebControls.DataList** when their **RepeatLayout** property is  
21 **Flow** . Furthermore, only unit types of **Pixel** and **Percentage** will work.

22 AddAttributesToRender

23  
24 [C#] protected virtual void AddAttributesToRender(HtmlTextWriter writer);

25 [C++] protected: virtual void AddAttributesToRender(HtmlTextWriter\* writer);

1 [VB] Overridable Protected Sub AddAttributesToRender(ByVal writer As  
2 HtmlTextWriter)  
3 [JScript] protected function AddAttributesToRender(writer : HtmlTextWriter);  
4

5 *Description*

6 Adds to the specified writer those HTML attributes and styles that need to  
7 be rendered. This method is primarily used by control developers. The output  
8 stream that renders HTML content to the client.

9 ApplyStyle

10  
11 [C#] public void ApplyStyle(Style s);  
12 [C++] public: void ApplyStyle(Style\* s);  
13 [VB] Public Sub ApplyStyle(ByVal s As Style)  
14 [JScript] public function ApplyStyle(s : Style);  
15

16 *Description*

17 Copies any non-blank elements of the specified style to the Web control,  
18 overwriting any existing style elements of the control. This method is primarily  
19 used by control developers. The style to be copied.

20 CopyBaseAttributes

21  
22 [C#] public void CopyBaseAttributes(WebControl controlSrc);  
23 [C++] public: void CopyBaseAttributes(WebControl\* controlSrc);  
24 [VB] Public Sub CopyBaseAttributes(ByVal controlSrc As WebControl)  
25 [JScript] public function CopyBaseAttributes(controlSrc : WebControl);

1  
2 *Description*

3 Copies the **System.Web.UI.WebControls.WebControl.AccessKey** ,  
4 **System.Web.UI.WebControls.WebControl.Enabled** ,  
5 **System.Web.UI.WebControls.WebControl.ToolTip** ,  
6 **System.Web.UI.WebControls.WebControl.TabIndex** , and  
7 **System.Web.UI.WebControls.WebControl.Attributes** properties onto the Web  
8 control from the specified source control. The source control with properties to be  
9 copied onto the Web control.

10 **CreateControlStyle**

11  
12 [C#] protected virtual Style CreateControlStyle();  
13 [C++] protected: virtual Style\* CreateControlStyle();  
14 [VB] Overridable Protected Function CreateControlStyle() As Style  
15 [JScript] protected function CreateControlStyle() : Style;  
16

17 *Description*

18 A protected method. Creates the style object that is used internally to  
19 implement all style-related properties. Controls may override to create an  
20 appropriately typed style.

21 *Return Value:* A **System.Web.UI.WebControls.Style** that is used to implement  
22 all style-related properties of the control.

23 **LoadViewState**

24  
25 [C#] protected override void LoadViewState(object savedState);



1 [C++] protected: void LoadViewState(Object\* savedState);

2 [VB] Overrides Protected Sub LoadViewState(ByVal savedState As Object)

3 [JScript] protected override function LoadViewState(savedState : Object);

4  
5 *Description*

6 Loads previously saved state. Overridden to handle ViewState, Style, and  
7 Attributes. Previously saved state.

8 MergeStyle

9  
10 [C#] public void MergeStyle(Style s);

11 [C++] public: void MergeStyle(Style\* s);

12 [VB] Public Sub MergeStyle(ByVal s As Style)

13 [JScript] public function MergeStyle(s : Style);

14  
15 *Description*

16 Copies any non-blank elements of the specified style to the Web control,  
17 but will not overwrite any existing style elements of the control. This method is  
18 primarily used by control developers. The style to be copied.

19 Render

20  
21 [C#] protected override void Render(HtmlTextWriter writer);

22 [C++] protected: void Render(HtmlTextWriter\* writer);

23 [VB] Overrides Protected Sub Render(ByVal writer As HtmlTextWriter)

24 [JScript] protected override function Render(writer : HtmlTextWriter);

1  
2 *Description*

3       Renders the control into the specified writer. The output stream that renders  
4 HTML content to the client.

5       RenderBeginTag

6  
7 [C#] public virtual void RenderBeginTag(HtmlTextWriter writer);  
8 [C++] public: virtual void RenderBeginTag(HtmlTextWriter\* writer);  
9 [VB] Overridable Public Sub RenderBeginTag(ByVal writer As HtmlTextWriter)  
10 [JScript] public function RenderBeginTag(writer : HtmlTextWriter);  
11

12 *Description*

13       Renders the HTML begin tag of the control into the specified writer. This  
14 method is primarily used by control developers.

15       This is made public so other controls can render multiple controls in  
16 between the begin tag and the end tag. The output stream that renders HTML  
17 content to the client.

18       RenderContents

19  
20 [C#] protected virtual void RenderContents(HtmlTextWriter writer);  
21 [C++] protected: virtual void RenderContents(HtmlTextWriter\* writer);  
22 [VB] Overridable Protected Sub RenderContents(ByVal writer As  
23 HtmlTextWriter)  
24 [JScript] protected function RenderContents(writer : HtmlTextWriter);  
25

1  
2 *Description*

3       Renders the contents of the control into the specified writer. This method is  
4 primarily used by control developers. The output stream that renders HTML  
5 content to the client.

6       RenderEndTag

7  
8 [C#] public virtual void RenderEndTag(HtmlTextWriter writer);  
9 [C++] public: virtual void RenderEndTag(HtmlTextWriter\* writer);  
10 [VB] Overridable Public Sub RenderEndTag(ByVal writer As HtmlTextWriter)  
11 [JScript] public function RenderEndTag(writer : HtmlTextWriter);  
12

13 *Description*

14       Renders the HTML end tag of the control into the specified writer. This  
15 method is primarily used by control developers.

16       This is made public so other controls can render multiple controls in  
17 between the begin tag and the end tag. The output stream that renders HTML  
18 content to the client.

19       SaveViewState

20  
21 [C#] protected override object SaveViewState();  
22 [C++] protected: Object\* SaveViewState();  
23 [VB] Overrides Protected Function SaveViewState() As Object  
24 [JScript] protected override function SaveViewState() : Object;  
25

1  
2 *Description*

3 A protected method. Saves any state that was modified after the  
4 **System.Web.UI.WebControls.Style.TrackViewState** method was invoked.

5 IAttributeAccessor.GetAttribute

6  
7 [C#] string IAttributeAccessor.GetAttribute(string name);

8 [C++] String\* IAttributeAccessor::GetAttribute(String\* name);

9 [VB] Function GetAttribute(ByVal name As String) As String Implements

10 IAttributeAccessor.GetAttribute

11 [JScript] function IAttributeAccessor.GetAttribute(name : String) : String;

12 IAttributeAccessor.SetAttribute

13  
14 [C#] void IAttributeAccessor.SetAttribute(string name, string value);

15 [C++] void IAttributeAccessor::SetAttribute(String\* name, String\* value);

16 [VB] Sub SetAttribute(ByVal name As String, ByVal value As String)

17 Implements IAttributeAccessor.SetAttribute

18 [JScript] function IAttributeAccessor.SetAttribute(name : String, value : String);

19 TrackViewState

20  
21 [C#] protected override void TrackViewState();

22 [C++] protected: void TrackViewState();

23 [VB] Overrides Protected Sub TrackViewState()

24 [JScript] protected override function TrackViewState();

1  
2 *Description*

3 Marks the beginning for tracking state changes on the control. Any changes  
4 made after "mark" will be tracked and saved as part of the control viewstate.

5 Xml class (System.Web.UI.WebControls)

6 TrackViewState

7  
8  
9 *Description*

10 Displays the contents of an XML document or the results of an XSL  
11 Transform.

12 Use the **System.Web.UI.WebControls.Xml** control to display the contents  
13 of an XML document or the results of an XSL Transform.

14 Xml

15 *Example Syntax:*

16 TrackViewState

17  
18 [C#] public Xml();

19 [C++] public: Xml();

20 [VB] Public Sub New()

21 [JScript] public function Xml();

22 ChildControlsCreated

23 ClientID

24 Context

25 Controls

1 Document  
2 TrackViewState

3  
4  
5 *Description*

6 Gets or sets the **System.Xml.XmlDocument** object to display.

7 DocumentContent

8 TrackViewState

9  
10 [C#] public string DocumentContent {get; set;}

11 [C++] public: \_\_property String\* get\_DocumentContent();public: \_\_property void  
12 set\_DocumentContent(String\*);

13 [VB] Public Property DocumentContent As String

14 [JScript] public function get DocumentContent() : String;public function set  
15 DocumentContent(String);

16  
17 *Description*

18 Gets or sets a string that contains the XML document to display in the  
19 **System.Web.UI.WebControls.Xml** control.

20 DocumentSource

21 TrackViewState

22  
23 [C#] public string DocumentSource {get; set;}

24 [C++] public: \_\_property String\* get\_DocumentSource();public: \_\_property void  
25 set\_DocumentSource(String\*);

1 [VB] Public Property DocumentSource As String

2 [JScript] public function get DocumentSource() : String;public function set

3 DocumentSource(String);

4  
5 *Description*

6 Gets or sets the URL to an XML document to display in the  
7 **System.Web.UI.WebControls.Xml** control.

8 EnableViewState

9 Events

10 HasChildViewState

11 ID

12 IsTrackingViewState

13 NamingContainer

14 Page

15 Parent

16 Site

17 TemplateSourceDirectory

18 Transform

19 TrackViewState

20  
21  
22 *Description*

23 Gets or sets the **System.Xml.Xsl.XslTransform** object that formats the  
24 XML document before it is written to the output stream.

25 TransformArgumentList

TrackViewState

[C#] public XsltArgumentList TransformArgumentList {get; set;}

[C++] public: \_\_property XsltArgumentList\*

get\_TransformArgumentList();public: \_\_property void

set\_TransformArgumentList(XsltArgumentList\*);

[VB] Public Property TransformArgumentList As XsltArgumentList

[JScript] public function get TransformArgumentList() : XsltArgumentList;public

function set TransformArgumentList(XsltArgumentList);

*Description*

TransformSource

TrackViewState

[C#] public string TransformSource {get; set;}

[C++] public: \_\_property String\* get\_TransformSource();public: \_\_property void

set\_TransformSource(String\*);

[VB] Public Property TransformSource As String

[JScript] public function get TransformSource() : String;public function set

TransformSource(String);

*Description*

Gets or sets the URL to an XSL Transform document that formats the XML document before it is written to the output stream.



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

UniqueID  
ViewState  
ViewStateIgnoresCase  
Visible  
AddParsedSubObject

[C#] protected override void AddParsedSubObject(object obj);  
[C++] protected: void AddParsedSubObject(Object\* obj);  
[VB] Overrides Protected Sub AddParsedSubObject(ByVal obj As Object)  
[JScript] protected override function AddParsedSubObject(obj : Object);

*Description*

Render

[C#] protected override void Render(HtmlTextWriter output);  
[C++] protected: void Render(HtmlTextWriter\* output);  
[VB] Overrides Protected Sub Render(ByVal output As HtmlTextWriter)  
[JScript] protected override function Render(output : HtmlTextWriter);

*Description*

Renders the results to the o

## EXEMPLARY COMPUTING SYSTEM AND ENVIRONMENT

Fig. 4 illustrates an example of a suitable computing environment 400 within which the programming framework 132 may be implemented (either fully or partially). The computing environment 400 may be utilized in the computer and network architectures described herein.

The exemplary computing environment 400 is only one example of a computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the computer and network architectures. Neither should the computing environment 400 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary computing environment 400.

The framework 132 may be implemented with numerous other general purpose or special purpose computing system environments or configurations. Examples of well known computing systems, environments, and/or configurations that may be suitable for use include, but are not limited to, personal computers, server computers, multiprocessor systems, microprocessor-based systems, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and so on. Compact or subset versions of the framework may also be implemented in clients of limited resources, such as cellular phones, personal digital assistants, handheld computers, or other communication/computing devices.

The framework 132 may be described in the general context of computer-executable instructions, such as program modules, being executed by one or more computers or other devices. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks

1 or implement particular abstract data types. The framework 132 may also be  
2 practiced in distributed computing environments where tasks are performed by  
3 remote processing devices that are linked through a communications network. In  
4 a distributed computing environment, program modules may be located in both  
5 local and remote computer storage media including memory storage devices.

6 The computing environment 400 includes a general-purpose computing  
7 device in the form of a computer 402. The components of computer 402 can  
8 include, by are not limited to, one or more processors or processing units 404, a  
9 system memory 406, and a system bus 408 that couples various system  
10 components including the processor 404 to the system memory 406.

11 The system bus 408 represents one or more of several possible types of bus  
12 structures, including a memory bus or memory controller, a peripheral bus, an  
13 accelerated graphics port, and a processor or local bus using any of a variety of  
14 bus architectures. By way of example, such architectures can include an Industry  
15 Standard Architecture (ISA) bus, a Micro Channel Architecture (MCA) bus, an  
16 Enhanced ISA (EISA) bus, a Video Electronics Standards Association (VESA)  
17 local bus, and a Peripheral Component Interconnects (PCI) bus also known as a  
18 Mezzanine bus.

19 Computer 402 typically includes a variety of computer readable media.  
20 Such media can be any available media that is accessible by computer 402 and  
21 includes both volatile and non-volatile media, removable and non-removable  
22 media.

23 The system memory 406 includes computer readable media in the form of  
24 volatile memory, such as random access memory (RAM) 410, and/or non-volatile  
25 memory, such as read only memory (ROM) 412. A basic input/output system

1 (BIOS) 414, containing the basic routines that help to transfer information  
2 between elements within computer 402, such as during start-up, is stored in ROM  
3 412. RAM 410 typically contains data and/or program modules that are  
4 immediately accessible to and/or presently operated on by the processing unit 404.

5 Computer 402 may also include other removable/non-removable,  
6 volatile/non-volatile computer storage media. By way of example, Fig. 4  
7 illustrates a hard disk drive 416 for reading from and writing to a non-removable,  
8 non-volatile magnetic media (not shown), a magnetic disk drive 418 for reading  
9 from and writing to a removable, non-volatile magnetic disk 420 (e.g., a "floppy  
10 disk"), and an optical disk drive 422 for reading from and/or writing to a  
11 removable, non-volatile optical disk 424 such as a CD-ROM, DVD-ROM, or other  
12 optical media. The hard disk drive 416, magnetic disk drive 418, and optical disk  
13 drive 422 are each connected to the system bus 408 by one or more data media  
14 interfaces 426. Alternatively, the hard disk drive 416, magnetic disk drive 418,  
15 and optical disk drive 422 can be connected to the system bus 408 by one or more  
16 interfaces (not shown).

17 The disk drives and their associated computer-readable media provide non-  
18 volatile storage of computer readable instructions, data structures, program  
19 modules, and other data for computer 402. Although the example illustrates a  
20 hard disk 416, a removable magnetic disk 420, and a removable optical disk 424,  
21 it is to be appreciated that other types of computer readable media which can store  
22 data that is accessible by a computer, such as magnetic cassettes or other magnetic  
23 storage devices, flash memory cards, CD-ROM, digital versatile disks (DVD) or  
24 other optical storage, random access memories (RAM), read only memories  
25 (ROM), electrically erasable programmable read-only memory (EEPROM), and

1 the like, can also be utilized to implement the exemplary computing system and  
2 environment.

3 Any number of program modules can be stored on the hard disk 416,  
4 magnetic disk 420, optical disk 424, ROM 412, and/or RAM 410, including by  
5 way of example, an operating system 426, one or more application programs 428,  
6 other program modules 430, and program data 432. Each of the operating system  
7 426, one or more application programs 428, other program modules 430, and  
8 program data 432 (or some combination thereof) may include elements of the  
9 programming framework 132.

10 A user can enter commands and information into computer 402 via input  
11 devices such as a keyboard 434 and a pointing device 436 (e.g., a "mouse").  
12 Other input devices 438 (not shown specifically) may include a microphone,  
13 joystick, game pad, satellite dish, serial port, scanner, and/or the like. These and  
14 other input devices are connected to the processing unit 404 via input/output  
15 interfaces 440 that are coupled to the system bus 408, but may be connected by  
16 other interface and bus structures, such as a parallel port, game port, or a universal  
17 serial bus (USB).

18 A monitor 442 or other type of display device can also be connected to the  
19 system bus 408 via an interface, such as a video adapter 444. In addition to the  
20 monitor 442, other output peripheral devices can include components such as  
21 speakers (not shown) and a printer 446 which can be connected to computer 402  
22 via the input/output interfaces 440.

23 Computer 402 can operate in a networked environment using logical  
24 connections to one or more remote computers, such as a remote computing device  
25 448. By way of example, the remote computing device 448 can be a personal

1 computer, portable computer, a server, a router, a network computer, a peer device  
2 or other common network node, and so on. The remote computing device 448 is  
3 illustrated as a portable computer that can include many or all of the elements and  
4 features described herein relative to computer 402.

5 Logical connections between computer 402 and the remote computer 448  
6 are depicted as a local area network (LAN) 450 and a general wide area network  
7 (WAN) 452. Such networking environments are commonplace in offices,  
8 enterprise-wide computer networks, intranets, and the Internet.

9 When implemented in a LAN networking environment, the computer 402 is  
10 connected to a local network 450 via a network interface or adapter 454. When  
11 implemented in a WAN networking environment, the computer 402 typically  
12 includes a modem 456 or other means for establishing communications over the  
13 wide network 452. The modem 456, which can be internal or external to computer  
14 402, can be connected to the system bus 408 via the input/output interfaces 440 or  
15 other appropriate mechanisms. It is to be appreciated that the illustrated network  
16 connections are exemplary and that other means of establishing communication  
17 link(s) between the computers 402 and 448 can be employed.

18 In a networked environment, such as that illustrated with computing  
19 environment 400, program modules depicted relative to the computer 402, or  
20 portions thereof, may be stored in a remote memory storage device. By way of  
21 example, remote application programs 458 reside on a memory device of remote  
22 computer 448. For purposes of illustration, application programs and other  
23 executable program components such as the operating system are illustrated herein  
24 as discrete blocks, although it is recognized that such programs and components  
25

1 reside at various times in different storage components of the computing device  
2 402, and are executed by the data processor(s) of the computer.

3 An implementation of the framework 132, and particularly, the API 142 or  
4 calls made to the API 142, may be stored on or transmitted across some form of  
5 computer readable media. Computer readable media can be any available media  
6 that can be accessed by a computer. By way of example, and not limitation,  
7 computer readable media may comprise "computer storage media" and  
8 "communications media." "Computer storage media" include volatile and non-  
9 volatile, removable and non-removable media implemented in any method or  
10 technology for storage of information such as computer readable instructions, data  
11 structures, program modules, or other data. Computer storage media includes, but  
12 is not limited to, RAM, ROM, EEPROM, flash memory or other memory  
13 technology, CD-ROM, digital versatile disks (DVD) or other optical storage,  
14 magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage  
15 devices, or any other medium which can be used to store the desired information  
16 and which can be accessed by a computer.

17 "Communication media" typically embodies computer readable  
18 instructions, data structures, program modules, or other data in a modulated data  
19 signal, such as carrier wave or other transport mechanism. Communication media  
20 also includes any information delivery media. The term "modulated data signal"  
21 means a signal that has one or more of its characteristics set or changed in such a  
22 manner as to encode information in the signal. By way of example, and not  
23 limitation, communication media includes wired media such as a wired network or  
24 direct-wired connection, and wireless media such as acoustic, RF, infrared, and  
25

1 other wireless media. Combinations of any of the above are also included within  
2 the scope of computer readable media.

3 Alternatively, portions of the framework may be implemented in hardware  
4 or a combination of hardware, software, and/or firmware. For example, one or  
5 more application specific integrated circuits (ASICs) or programmable logic  
6 devices (PLDs) could be designed or programmed to implement one or more  
7 portions of the framework.

### 8 9 Conclusion

10 Although the invention has been described in language specific to structural  
11 features and/or methodological acts, it is to be understood that the invention  
12 defined in the appended claims is not necessarily limited to the specific features or  
13 acts described. Rather, the specific features and acts are disclosed as exemplary  
14 forms of implementing the claimed invention.